B26

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/30

A1

(11) International Publication Number: WO 99/52045

(43) International Publication Date: 14 October 1999 (14.10.99)

(21) International Application Number: PCT/US99/06724

(22) International Filing Date: 29 March 1999 (29.03.99)

(30) Priority Data:
09/054,861 3 April 1998 (03.04.98) US

(71) Applicant: AVID TECHNOLOGY, INC. [US/US]; Metropolitan Technology Park, One Park West, Tewksbury, MA 01876 (US).

(72) Inventor: VALDEZ, Luis, J., Jr.; 773 Tremont Street #1, Boston, MA 02118 (US).

(74) Agent: GORDON, Peter, J.; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210 (US).

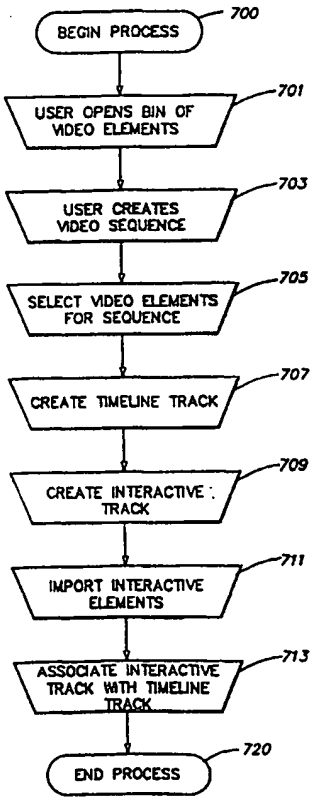
(81) Designated States: AU, CA, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: SYSTEM AND METHOD FOR PROVIDING INTERACTIVE COMPONENTS IN MOTION VIDEO

(57) Abstract

An editing system and delivery system synchronizes the transmission of interactive elements with a video signal, such that the interactive components may supplement information provided in the video signal at predefined periods. The interactive elements are transmitted in relation to the video such that the interactive elements will arrive in time to display the components concurrently with a portion of the video presentation. The interactive elements may be transmitted within a video signal to a viewer. The viewer is configured to display the interactive elements in response to a signal from the delivery system. An editing system is disclosed that may define a programming track for interactive elements that may be associated with one or more audio and video tracks. The editing system schedules transmission and appearance of the elements. A graphical user interface of the editing system may display a time in a presentation timeline when the interactive elements appear and the duration which the interactive elements are displayed. Graphical user interface representations of the interactive elements are translated into commands including file transfer instructions to be executed on a playback system. The playback system may transmit information element data within a video signal to a viewer.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

SYSTEM AND METHOD FOR PROVIDING INTERACTIVE COMPONENTS IN MOTION VIDEO

FIELD OF THE INVENTION

The present invention relates generally to a system and method for providing interactive components, such as markup language elements in motion video.

BACKGROUND OF THE INVENTION

Generally, computer systems exist for editing, playing, and broadcasting motion video. One such editing and playback system is the Media Composer video production system provided by Avid Technology, Inc. (Media Composer is a registered trademark of Avid Technology, Inc.). These computer systems typically digitize one or more video and audio tracks and store them on computer storage media, such as a hard disk device. Software systems may then modify, combine, or enhance these video and audio tracks to produce a broadcast-quality production, such as a motion picture or newscast. These systems may include editing software that may allow a user to select different segments of audio and video, prepare clips for replay, and perform splicing between clips, among other functions.

Computer systems also exist that are capable of displaying interactive documents having interactive elements. In particular, markup languages allow a user to define a number of different types of elements within an interactive document, some of which are capable of handling interactions with a user. Interactions may include selecting an interactive element within the displayed electronic document, using a pointing device such as a mouse. Such elements may include a predefined area of a still picture or a series of text. A user may also input information into a displayed element, using the keyboard or other input device. When a user interacts with the document, the interactions may cause additional information to be displayed to the user.

Examples of markup languages generally used to produce interactive electronic documents include SGML, XML, HTML, and Dynamic HTML, among others. The Standard Generalized Markup Language ("SGML") is used to represent a wide variety of document types such as books, electronic software documentation, and equipment specifications, among other applications. SGML is an international standard (ISO-8879) published in 1986 for the electronic publication of documents. SGML defines a markup language wherein content of a

document is structured using markup, i.e., tags or codes encapsulating the content. The markup defines elements which form a logical, predictable structure. SGML defines a strict markup scheme with a syntax for defining document elements and an overall framework for marking up documents. A document type definition (DTD) of SGML establishes the structure of a markup document of a particular type, and provides a framework for the kinds of elements that constitute a document of that type. The markup of a document is interpreted as an ordered hierarchy of markup elements which when, taken together, form a tree or similar hierarchial object. A markup element describes the function or meaning of the content which it includes.

In such a document, markup elements include tags and their content, such as text, graphics, still images or other media. A markup language document includes markup tags that may be described as start tags, end tags, or empty tags. A start tag begins a markup element. An end tag ends the corresponding markup element. These start tags and end tags define the element in markup languages, such as a book, library, or body of a document. An empty tag is understood as being both a start tag and an end tag with no content between the start and end tags. Between a start tag and an end tag other start tags and corresponding end tags may be arranged in a hierarchial manner such that there are children elements and parent elements having a defined relationship to each other. These elements may define a series of interactive elements that may receive user input. Such elements may include buttons, menus, text fields, graphics, links to other documents or other interactive markup elements as defined by the markup language. When a user provides input to these elements through a document viewer/browser interface, the viewer may respond by modifying, transmitting, or receiving displayed or stored data.

Also in markup language documents, there are elements that contain metadata, or information about the document. Metadata may describe document information such as location, name, and creation date of an electronic document that may accompany the document or may be embedded in the document itself. Metadata is typically used to catalogue electronic documents or otherwise identify information relative to an electronic document.

Hypertext Markup Language (HTML), eXtensible Markup Language (XML) and Dynamic HTML define particular document types that conform to SGML by having a definitive DTD. HTML, XML, and Dynamic HTML are widely used over the Internet for distributing information between servers and clients. These markup language document types

may be edited, viewed and verified according to their respective DTDs. By distributing markup language documents through networks such as the Internet, information providers can make information available to a large number of consumers. These interactive documents are generally transferred between a "server", or provider system, and a "client", or consumer system over a communication network. In the Internet, documents may be transferred using a transfer protocol referred to in the art as the Hypertext Transfer Protocol (HTTP). This application-layer protocol uses a network-layer transport protocol such as the well-known Transmission Control Protocol/Internet Protocol (TCP/IP) to transfer these documents over the network between a server and a client. When transmitting information to many clients, a protocol vehicle referred to as multicast is used to transfer information economically. Multicast, TCP/IP, and HTTP are well-known in the art of computer communications protocols.

Documents may be distributed over various types of networks, such as Ethernet, Fast Ethernet, ATM, FDDI, dial-up networks, among others. Documents may also be broadcast or multicast to multiple users over a network such as a video broadcast network. Specifically, documents may be transmitted within a video signal, such within a vertical blanking interval signal (VBI) of a video transmission signal. The VBI is the section of the video signal that may be used for transmitting data, such as closed-captioning information. One such method of transmitting multimedia data over networks and within video signals is disclosed in the Broadcast-Enabled PC Software Development Kit provided by the Microsoft Corporation. The software development kit and its associated library functions provide a facility for transmitting multicast IP data to a number of clients over a variety of network interfaces.

Markup language documents are generally viewed using a software program referred to in the art as a browser or viewer. A browser interprets a series of elements of a markup language document as browser instructions. The elements contain text or images, and a number of formatting commands, when interpreted, change the appearance of text or images within the display generated by the browser. Browsers typically interpret interactive markup language elements and handle input and output events associated with these interactive elements. Examples of browsers include the Navigator and Microsoft Explorer browser programs provided by the Netscape Corporation and the Microsoft Corporation, respectively (Navigator is a registered trademark of the Netscape Corporation). These browsers are

software systems that are generally installed on a computer and computer user to enable the computer to view and edit markup language documents.

Another system for viewing markup language documents is the WebTV browser provided by WebTV Networks, Inc. and various other manufacturers (WebTV and WebTV Network are registered trademarks of WebTV Networks, Inc.). The WebTV Internet browser is a system that allows a user to browse markup language documents published on the Internet without the need for a computer. The WebTV Internet browser includes hardware connected to a standard television and phone line. The hardware obtains documents through the phone line using HTTP and TCP/IP protocols in a well-known manner. A recent improvement of the WebTV browser hardware, referred to as WebTV Plus hardware, allows a content provider to broadcast markup language documents. This improved WebTV hardware may display both video and markup language data on the screen at the same, to provide information such as stock information reports, channel guides, etc. The video information generally appears in one rectangular section of the screen reserved for displaying video in what is commonly referred to as "picture-in-picture" format, or in WebTV Plus, "Web PIP" format. This display of data in the rectangular section of the screen is independent of the display of data in other areas of the screen. The markup language document is displayed in a separate section of the screen and may receive interactive input from the user. A user may "flip" back and forth between the video information and markup language document that is displayed on the television screen.

SUMMARY OF THE INVENTION

Prior systems do not allow a user to associate interactive data with a video presentation in a time-based relationship over a distributed network. An editing system and delivery system is provided that synchronizes the transmission of interactive elements with a video signal, such that the interactive components may supplement information provided in the video signal at predefined periods. The interactive elements are transmitted in relation to the video such that the interactive elements will arrive in time to display the components concurrently with a portion of the video presentation. The interactive elements may be transmitted within a video signal to a viewer. The viewer is configured to display the interactive elements in response to a signal from the delivery system.

An editing system is disclosed that may define a programming track for interactive elements that may be associated with one or more audio and video tracks. The editing system

schedules transmission and appearance of the elements. A graphical user interface of the editing system may display a time in a presentation timeline when the interactive elements appear and the duration which the interactive elements are displayed. Graphical user interface representations of the interactive elements are translated into commands including file transfer instructions to be executed on a playback system. The playback system may transmit information element data within a video signal to a viewer.

This ability to synchronize interactive elements with portions of a video presentation provides an audience an enhanced presentation experience. Audience users may then interact with the presentation they are watching, because various interactive elements are displayed to the user when a particular portion of the video presentation is played. A user then may interact with the interactive elements and receive more information associated with the presentation. The ability to provide interactive elements with a video presentation may be particularly useful for video training and presentations, newscasts, entertainment programs, and other viewing environments where a heightened viewing experience is desired.

According to one aspect, a method is provided for processing a video signal using an editing system program. The method involves of associating an interactive element to a portion of a video presentation to be transmitted and displayed on a viewer, wherein the interactive element has a predefined relationship to the portion and transmitting the interactive element to the viewer at a predetermined time relative to a transmission of the portion.

According to another aspect, a system and method is provided that determines a time relation between a video broadcast and an interactive video element. An interactive element is inserted into the video signal based on the time relation.

According to another aspect, method for transmitting a video signal is provided that receives a composition of a multimedia presentation, the composition including one or more interactive elements having a time relation to one or more video elements. The interactive elements are encoded within the video signal and the video signal is transmitted.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings,

Fig. 1 is a block diagram of an exemplary computer system and network that may be suitable for implementing various exemplary embodiments;

Fig. 2 is a block diagram depicting an editing and delivery system suitable for implementing various embodiments;

Fig. 3A is a block diagram that shows one embodiment of an editing and delivery system;

Figs. 3B - 3D show various embodiments of an editing and delivery system;

Fig. 4 is an object-oriented diagram showing a class hierarchy of object classes of objects that may be created in an editing system;

Fig. 5 is a diagram illustrating a hierarchy of a composition;

Fig. 6A is an example display of an interactive interface that controls an interactive programming track;

Figs. 6B - 6C are example display views produced by the editing system of Figs. 2 and 3A-3D;

Fig. 7 is a flow chart showing a method of processing an interactive element in an editing system;

Fig. 8 is a flow chart showing a process for determining bandwidth requirements;

Fig. 9 is a flow chart showing a process of associating elements; and

Fig. 10 is a flow chart showing a process for transmitting video and element data.

DETAILED DESCRIPTION

The present invention will be more completely understood through the following detailed description which should be read in conjunction with the attached drawing in which similar reference numbers indicate similar structures. All references cited herein are hereby expressly incorporated by reference.

An example computer system for implementing this system is depicted in Fig. 1. The computer system 101 includes a processor 104 having one or more storage devices 108, such as one of more disk drives. The computer system also includes one or more output devices 107, such as monitors or graphic displays (not shown), or printing devices (not shown). The computer system 101 typically includes a memory 105 for storing programs and data during operation of the computer system 101. In addition, the computer system may contain one or more communication devices 109 that connect to a communication network 111. Computer system 101 may also include a video data I/O subsystem 110 that can convert analog audio and video data into digital information. Subsystem 110 may also be capable of accepting

digitized audio and video information directly for use by system 101. It should be understood that this system is merely one system that may be used, and thus the present invention is not limited to the specific systems described herein.

Computer system 101 may be a general purpose computer system, that is programmable using a high level computer programming language. The computer system may also be implemented using specially programmed, special purpose hardware. In the computer system 101, the processor 104 is typically a commercially available processor, such as the PENTIUM microprocessor from the Intel Corporation, PowerPC microprocessor, SPARC processor, PA-RISC processor, M68000 series microprocessor or MIPS processor. Many other processors are available. Such a processor usually executes an operating system which may be, for example, the DOS, WINDOWS 95, WINDOWS 98, WINDOWS NT, SYSTEM 7, SOLARIS, NetWare, InternetWare, IRIX, or UNIX operating system (WINDOWS is a registered trademark of the Microsoft Corporation, SYSTEM 7 is a registered trademark of the Apple Corporation, Solaris is a registered trademark of Sun Microsystems, Inc., IRIX is a trademark of the Silicon Graphics Corporation, MIPS is a registered trademark of MIPS Technologies, Inc., and NetWare and InternetWare are registered trademarks of the Novell Corporation.)

The communication network 111 may be an Ethernet or other type of local or wide area network (LAN or WAN), a point-to-point network provided by telephone services, or other type of communication network. Information consumers and providers referred to as server 102 and client 103 systems, respectively, communicate through the network 111 to exchange information. Computer system 101 may be configured to perform as a client 102 or server 103 system or both on the network 111. A server such as 103A may store markup language documents and/or video files on a storage device 108 located on the server though not shown. The server may also provide these documents and files to one or more client systems 102 in response to a request generated by a client such as 102A through the network 111. Similarly, markup language, video and audio files documents according to the present invention may be created, edited, viewed, or arranged on such client 102 and server 103 systems.

It should be understood that the invention is not limited to a particular computer system platform, processor, operating system, or network. Also, it should be apparent to those skilled in the art that the present invention is not limited to a specific programming language

or computer system and that other appropriate programming languages and other appropriate computer systems could also be used.

Having now described an exemplary computer system on which this system may be implemented, an editing and delivery system will now be described in more detail in connection with Figs. 2 and 3A-3D.

Fig. 2 shows one exemplary embodiment of an editing and delivery system 200. The system 200 may include an editing system 201 that may edit video and document presentations for distribution to a number of viewer systems 215. Editing system 201 may use various audio and video media files stored on storage system 209 to create a composition. Editing system 201 may be capable of handling one or more tracks of audio/video information, and may be capable of performing editing functions such as dissolves, wipes, flips, flops, and other functions known in the art of video production. These media files are typically created by a digitizing system 205 that receives one or more audio/video inputs from media player 207. These media files may also be digitized directly by a digital recorder (not shown). Editing system 201 may also use interactive elements in creating a composition.

These interactive elements may take the form of markup language documents having interactive elements that process inputs from a user or other system. Markup language documents may be sorted on storage system 209, or in any other system on any network 103. Network 103 may be, for example, an Ethernet, Fast Ethernet, ATM, or FDDI network, or any other network type known now or in the future. Network 103 may also be a communication bus within a computer or any specially-developed communication hardware. Playback system 211 may distribute these compositions to a number of viewers 215 by transmitting composition data to viewers 215 through network 203, broadcast transmission 213, or any other method for distributing data. Broadcast transmission 213 may be, for example, a transmission of a video signal such as through a cable television broadcast signal, satellite broadcast, or the like. Viewer systems 215A and 215B are consumers, or clients, of the content of the composition. Through these viewers 215, a user or system may interact with the composition and its interactive elements.

It should be understood that one or more parts of system 200 may be implemented on one or more systems. Further, it should be understood that the system may be embodied as a single system having no separable components.

Such an editing system 201 suitable for implementing the present invention is described in one or more U.S. patent applications, including U.S. Patent Application Serial No. 08/465,812 filed on June 6, 1995 by Michael J. Wissner entitled METHOD AND APPARATUS FOR REPRESENTING AND EDITING MULTIMEDIA COMPOSITIONS USING REFERENCES TO TRACKS IN THE COMPOSITION TO DEFINE COMPONENTS OF THE COMPOSITION, incorporated by reference. Such an editing system and its interface include U.S. Patent Application Serial No. 08/689,577 filed on August 12, 1996 by Phillip R. Moorby et al. entitled MULTIMEDIA VISUAL LANGUAGE METHOD AND APPARATUS, incorporated herein by reference. An example of storage system 209 is described in U.S. Patent No. 5,267,351, filed on Dec. 22, 1989 by Stephen J. Reber et al. entitled MEDIA STORAGE AND RETRIEVAL SYSTEM, incorporated herein by reference. An example of playback system 211 suitable for implementing the present invention is described in U.S. Patent No. 5,045,940, filed Dec. 22, 1989 by Eric C. Peters entitled VIDEO/AUDIO TRANSMISSION SYSTEM AND METHOD, incorporated herein by reference. An example of a suitable format for media compositions and elements of a composition includes the Open Media Framework (OMF) Interchange format described in the OMF Interchange Specification, Version 2.1, dated September 18, 1997, incorporated herein by reference. Another suitable media file format may include the Advanced Authoring Format (AAF) developed by a number of companies participating through the Multimedia Task Force (MMTF), incorporated herein by reference.

Further, a commercial editing system 201 suitable for implementing the present invention may be used, such as the Media Composer video production system or NewsCutter news editing system available from Avid Technology, Inc. (NewsCutter is a registered trademark of Avid Technology, Inc.). Also, a commercial playback system 211 suitable for implementing the present invention may be used that implements the Media Engine video playback system available from Avid Technology, Inc. that is incorporated in the Avid AirPlay MP playback server system (AirPlay is a registered trademark of Avid Technology, Inc.). A commercial storage system 209 suitable for storing composition files includes the MediaShare external storage device (MediaShare is a trademark of Avid Technology, Inc.). Other commercial systems may be used.

One or more of these systems may be adapted to associate interactive elements with a composition to create an interactive programming presentation. Interactive programming is a

term referring to an enhanced television and video experience that includes the transmission of auxiliary data to display text, graphics and other forms of media concurrently with a conventional television signal. A user may be able to interact with the data in the interactive programming presentation.

Having described systems that may be modified to edit and deliver interactive programming presentations, various embodiments of editing system and playback systems will now be described with reference to Figs. 3A-3D.

Figures 3A-3D show various illustrative embodiments of a client-server system used to associate interactive elements with a composition, and a delivery system for an interactive programming presentation. Fig. 3A shows an editing, playback and storage system 201, 209, 211 and its components. It should be understood that one or more of these functions may be distributed or incorporated in the same system, and is shown in Fig. 3A within a single system for simplicity. System 201, 209, 211 includes media storage 313 and media database 315, the latter being used as a locator facility for locating media files within media storage 313. Media editor 309 provides a user interface and facilities for editing a composition of media objects.

The media objects may be, for example, complex objects that refer to media data objects in raw data format and include information relative to the media data objects. These objects will be referred to hereinafter as composition media objects. A composition media object contains information for playing a piece of media represented by a media data object. The composition media object generally includes no actual data; rather, the object includes descriptive information that embodies a media object.

Media objects used in a composition may include source material such as video or audio tape, compact disc, computer generated images, and the like. Currently available digital representations of source material are referred to herein as source media objects. Source media objects contain digital data which correspond to original source material, as well as information regarding how media data objects were created, and an identification of the corresponding source material. Thus, a source media object may contain a source content identifier, identifying the original source material.

A source media object may also include a sample rate in the link of each sample and bytes. The media object also includes an indication of the section of original source material that it represents. The object stores the time offset from the source origin of its first sample. The unit of this offset is the sample duration for the media data file. A media data file may

also include fields containing information regarding the content of the media, the quality of the media, or other information. Media objects may assume different formats such as a run-link encoded (RLE) RGBA data format, JPEG format, file interchange format (JFIF) format, or the like. A typical graphics format used to create graphic images overlaid on the video includes RLE pixel array, TIFF, or the like. A suitable format for audio data is the audio interchange file format (AIFF), WAVE audio file format, among others. For text files, such as commentary, file names, and other text associated with other media, a media object may include text in various formats including ASCII or any other character encoding.

To support editing of compositions of such a variety of media, a media editing system 309 is provided that may create data structures for organizing and storing information regarding a composition and perform operations for manipulating these data structures. Media editing system 309 stores various media objects within media storage 313, such as a source media object 325 and composition media object 327. Objects stored within media storage 313 may be generally accessed through a media database 315. Media database 315 contains references to individual objects stored on media storage 313. Media editing system 309 arranges these objects to form compositions that contain references to the source media objects 325. Thus, media editing system 309 manipulates objects that are references to actual data.

Media playback 311 provides a facility for playing back compositions locally at the playback system or may transmit a composition as video transmission 321 and data transmission 323. These transmissions are interpreted by a viewer 215 that is capable of displaying received video signals and interpreting interactive elements. Viewer 215 may be, for example, an Intel-processor based PC running the Microsoft Windows 98 Operating System an having television signal viewing hardware or a standard television having a WebTV hardware device attached. Viewer 215 may also have an additional data connection that would allow the viewer 215 to receive additional information from other sources, such as from the Internet.

Media playback 311 may also transmit interactive elements 329 over a typical data transmission 323 or embedded in another signal, such as a video transmission 321. Media playback 311 generally transmits data using one or more network protocols such as TCP/IP, or the like. Media playback 311 generally sends information to a network interface 303 through a series of drivers such as protocol driver 307 and media device driver 305. Network

interface 303 may be, for example, an Ethernet or other network interface, serial connection or modem. Other interfaces may be used. Protocol driver 307 is generally associated with a data link or network layer protocol driver whereas media device 305 is generally a driver associated with a particular network interface 303. Media device drivers and protocol drivers are generally known in the art of computer programming.

Media playback 311 may transmit a composition also in a video format for broadcast to a viewer 215 as a video stream. Video streams may generally be transmitted directly or broadcast through the air, via videotape, a cable, or similar medium. Media playback will generally transmit information to a video interface 317 using a video device driver 319. Video interfaces and video device drivers are well-known in the art of computer programming. Thus, a playback system 211 may transmit a composition over single or multiple data video transmission paths.

Figs. 3B-3D show various embodiments of a media playback system. In Fig. 3B, playback system 345 transmits both data and video over a video data transmission channel 335. A video and data decoder 337 is provided that may decode signal 335 into a separate data transmission 323 and video transmission 321. These separate transmissions may be then processed by associated hardware or software within viewer 215. In playback system 345, network interface 331 and video interface 333 may be incorporated in a single hardware solution. Fig. 3C shows a playback system 211 that transmits separate data and video transmissions. In addition, a video and data encoder 347 is provided which produces a video and data transmission 335 from a separate data transmission 323 and video transmission 321. The video and data encoder 347 may be, for example, a VBI insertion device for inserting data into one or more lines within the video blanking interval of video transmission 321. Fig. 3D shows an embodiment wherein both viewer and media playback process the video and data transmission 335 directly.

As discussed, TCP/IP information may be transmitted using the vertical blanking interval (VBI) of a television signal. The VBI is a non-viewable portion of the television signal that may be used to provide data services using a variety of protocols. One such use for the VBI includes the transmission of multicast TCP/IP data over point-to-point connections. Multicast methods of distributing data are very useful for the distribution of news feeds, stock quotes, news groups, weather reports, and the like. Generally, multicast communication is

useful when sending data to a large group of individual clients because only one copy of data needs to be transmitted by a sender to arrive at multiple clients.

The VBI may be included, for example, within a television signal such as an NTSC television frame. An NTSC television frame comprises two fields having 262.5 horizontal scan lines each. The first 21 lines of each field are not part of the visible picture and are collectively called the vertical blanking interval (VBI). Of these 21 lines, the first 9 are used while repositioning the cathode ray of the television to the top of the screen, but the remaining lines are available for data transport. Line 21 itself is reserved for the transport of closed captioning data. There are therefore eleven (11) possible VBI lines being broadcast 60 times per second (each field 30 times per second), some or all of which may be used for transmitting data. The availability of eleven lines of VBI yields approximately 115 kbps of available bandwidth. It should be understood that one or more of these lines may be used for existing proprietary data or testing services. Thus, transmission of data over the VBI may be just one data service using a subset of these lines.

The amount of available bandwidth for data transmission depends upon the number of lines used within the VBI signal to transmit data. The number of lines used may be adjusted, depending on the amount of bandwidth required.

The North American Basic Teletext Standard (NABTS) is defined in the Electronics Industry Associations Standard EIA-516. This standard provides an industry-accepted method of modulating data onto the VBI of an NTSC signal. Further information concerning the NABTS standard and its implementation may be found in standard EIA-516, incorporated herein by reference.

The NABTS packet is a 36-byte data structure encoded onto one horizontal scan line of an NTSC signal having the following structure:

[2-byte clock sync][1-byte sync][3-byte packet group address][1-bye continuity index][1-byte packet structure flags][26-byte data block][2-byte FEC suffix]

The 2-byte clock synchronization and 1-byte byte synchronization, although not part of the NABTS packet, are located at the beginning of every scan line containing an NABTS packet and are used to synchronize the decoding sampling rate and byte timing. The 3-byte packet group address field is Hamming encoded (as specified in EIA-516, and provides four

data bits per byte), and thus provides 4096 possible packet group addresses. These addresses are used to distinguish related services originating from the same source, allowing the receiver to determine which packets are related and part of the service. The 1-byte continuity index field is a Hamming encoded byte, which is incremented by one for each packet of a given packet group address. The index number is determined by the packet's order in the forward error correction (FEC) bundle mentioned below. The first packet in the bundle has a count 0, and the two FEC only packets at the end have counts 14 and 15, respectively. This index number allows the decoder to determine if packets have been lost during transmission.

The packet structure field is also a Hamming encoded byte, which contains information about the structure of the remaining portions of the packet. The most significant bit is 0 according to one embodiment. The second significant bit typically specifies whether the data block is full, that is, 0 indicates the data block is full of useful data, 1 indicates some or all of the data is filler data. The least two significant bits are used to indicate the length of the suffix on the data block, in this implementation, either 2 or 28 bytes. This suffix may be used for forward error correction described below.

The data block field may include 0 to 26 bytes of useful data. Filler data may be indicated by a 0x15 following by as many 0xEA as are needed to fill the packet. Sequential data blocks minus filler data form an asynchronous serial stream of data.

These NABTS packets are modulated onto the NTSC signal sequentially and on any combination of lines. Due to the unidirectional nature of VBI data transport, forward error correction (FEC) may be required to insure the integrity of data at the receiver. Any forward error correction could be used for this purpose. The FEC for NABTS is capable of correcting single byte errors and single and double byte erasures in a data block and suffix of an NABTS packet. Further information concerning the packet format and transmission of multicast TCP/IP information over the VBI is disclosed in Proposal-Vertical Blanking Interval (VBI) Standard to Link TV Networks to the Internet, an Internet Draft Proposal submitted to the Internet Engineering Task Force (IETF) and incorporated herein by reference.

A number of VBI encoder and decoder products are commercially available for the purposes described herein. Several devices for encoding information into a television signal include the TES3 and GES3 VBI encoder and /decoder units available from the Norpack Corporation. A WebTV hardware device may also be used as a decoder device. Other

encoder and decoder solutions may be used to decode transmitted VBI data. It is understood that other methods for transmitting data and video may be used.

Creating Compositions

The basic building blocks of a composition are called components. The composition is structured as a tree of components; it has a root component and a component may or may not have some subcomponents, depending on its type. A component is a function over time because it contains information for producing a state of its portion of the composition at any time within its range. A component thus represents a time-dependent sequence of media data objects or sources called a media stream.

The simplest elements of a composition are source components or "clips" representing sections of actual media such as video on a videotape, audio on a compact disk, computer-generated image on a computer disk, or the like. Other components represent different combinations of media streams which produce new media streams. Such components represent the variety of relationships created by editing operations performed, for example, by video editing systems and audio mixers.

Components may be categorized and implemented in several different ways. It may be preferable to use a form of object-oriented programming to enforce categorization. The above-described computer system may be programmed using an object-oriented programming language, such as C++, to provide definitions of types of components. Through object-oriented programming, components may inherit common functions between component types and may enable enforcement of structural rules for a composition.

In an object-oriented composition editing and representation system, there may be two types of hierarchies. The first type of hierarchy is abstract, illustrating how one component type may be a subtype of another component type. Such a hierarchy is known as a class hierarchy and will be described below in further detail in connection with Fig. 4. A second type of hierarchy is a structural hierarchy, that indicates how a composition is composed hierarchically of instances of abstract components. A structural hierarchy of a composition will be described in further detail with reference to Fig. 5.

Data structures used for representing a composition generally exclude media data, by containing indications of or references to the actual media data and representations of the relationships between the media that form the composition. Thus, positions are stored separately from the actual media data to which they refer, allowing many compositions to use

the same media data without duplicating it. With such a structure, a composition need not be reproduced when it is changed. Further, the composition itself may not actually reproduce the presentation but merely represents it and provides reference information for the playback of the presentation. As shown in Fig. 4, there may be several classes of components in a composition.

A class is a category of object such that all objects within that class are similar in representation and functionality. These classes may be implemented in a computer system using an object-oriented program construct referred to in the art as a class. Some of these classes are abstract classes, of which no components are direct members. Components may, however be indirect members of an abstract class by virtue of being direct members of a subclass of the abstract class. Because there are no direct members of an abstract class, the purpose of defining an abstract is to express the commonality of the subclasses of that class. It enables operations to be defined once for the abstract class rather than multiple times, once for each subclass. Classes that are not abstract, and therefore which may have direct members, are called concrete classes.

Specific classes will now be described in connection with Fig. 4. It should be understood that other classes may be implemented, and the invention is not limited to the specific classes shown.

The top level, abstract, class of a composition is referred to as a component 401, and defines functionality common to all components. For each component created for a composition, memory locations are allocated to store and group together relevant information concerning that component. A component that is a member of a subclass of a component class inherits characteristics of the component class. Information stored as part of each component to be described in more detail below includes:

1) Track Type
2) Player Function Code
3) Edit Rate
4) Parent
5) Subcomponent Identifier
6) Edit Nesting Level
7) Precompute
8) Name
9) Attribute List

Track type (1) provides an indication of the type of material, or media data, represented by the component, such as video, audio, etc. Player function code (2) indicates an algorithm for displaying material represented by the component.

A component may also include edit rate (3) that is a representation of time units used for determining the duration of the component. The edit rate differs from the actual sample durations stored in the media data, and may be thought of as a "virtual" sample rate. An edit rate is a number of edit units per second and is not limited to being an integer. For example, the edit rate could be a floating point decimal. The edit rate thus defines duration of virtual samples within a component, and determines the smallest editable unit of media. For example, a frame-based 30 frame per second NTSC video editing system may use an edit unit of 29.97 for its video components. An audio editing system for editing audio associated with the video may use the same edit rate of 29.97. Thus, media data may be substituted at a later time with media digitized at a different sample rate, and the composition may still be valid because enough information is maintained to perform a sample rate conversion. Sample rate conversion capabilities are well-known in the art.

A component may also contain a reference to a component which depends from it within the composition, this reference will be referred to as a parent pointer (4). The component may also contain a subcomponent identifier (5) that identifies a component in its parents context. Because a composition is hierarchical, every component, except the root component, has a parent component. By maintaining parent pointers it is possible, when a composition is edited, to find dependencies on a particular component, enabling a composition manager, for example, to readily locate components affected by a change to the composition.

A component may also contain an optional pointer (7), another component representing a precomputed media data file. A precomputed media data file is one that contains a concrete representation of an intermediate media stream produced by the component it is attached to. This feature enables an application to play a composition in real-time, when it would not be possible to compute in real time the media effects represented by the composition. Further, the feature enables an editing system to see the precomputed result, or to remake an effect from original source material.

The edit nesting level (6) may be used for identifying nested editing steps for editing a component. Its use will be described in more detail below in connection with the description of editing operations.

Name (8) and attribute list (9) of a component are examples of optional information for identifying contents of a component.

A component need not maintain any explicit indication of its location in time within a composition. This location may be determined by following parent pointer links to the root of the composition, and, at each level, passing subcomponent identifiers to the parent. The parent then computes the offset of the subcomponent within its own context and adds it to an accumulating offset. By not storing this information explicitly, the number of components which are required to be examined during an edit may be limited. Thus, high efficiency during editing is maintained.

Segment 403 and transition 402 are two subclasses of component class 401, thus these two subclasses inherit data and structure defined in component 401.

A segment is an abstract class of component 401 that represents an independent description of a section of a presentation. A segment is independent because it provides a meaningful representation of a portion of a composition even when viewed out of context of the rest of the composition. A segment may be a simple clip representing some kind of media, or it may be a more complex structure such as a sequence, or a track group defined in more detail below. A segment is also a representation for a track and inherits functionality from component 401. There are three subclasses of the segment class 403: clip 404, sequence 405 and track group 406.

A sequence, a concrete subclass of a segment, represents serialization or Concatenation in time of other subcomponents. The sequence, as its name implies, is an ordered list of segments separated by transitions, with the restriction that it begins and ends with a segment. This restriction allows sequences to be treated as segments, which may be included in other sequences.

The order of segments in a sequence defines the order of interpretation or "playback" thus, sequencing information is represented implicitedly by the structure of the sequence, rather than explicitly through the use of relative start times and durations for clips. The sequence may include the following information:

1) Ordered collection of subcomponents

A) Length of subcomponent in edit units

B) Subcomponent identifier (pointer)

A sequence of media data may be represented in the form of a tree of its subcomponents rather than linearly. A modification of a balanced binary tree data structure (a modified 2-3-tree) may be used to store subcomponents.

A clip 404 is an abstract class of component 401 that represents a single contiguous section of a piece of media, with a specified position in the media and a specified length. A clip may further contain information on its length, that explicitly determines its duration. The position of a clip in the source media may be explicit or implicit, depending on its subclass. There are three concrete subclasses of clips: Source references 407, track references 408, and user attributes 409.

Compositions may represent arbitrarily complex manipulations of media, but ultimately the compositions are based on physical sources, such as a video tape from a camera or graphics produced by a computer program. References to source material are embodied in source clips 407 that describe single time-contiguous sections of a source, by noting a position in the source media, and a length. A source reference 407 does not contain actual media but only references it. Source references 407 may represent the lowest level, or leaf components, of a composition. A source reference 407 inherits properties and functions of a clip, but also contains the following information:

1) Physical Rate
2) Physical Start
3) Physical Length
4) Source Identifier
5) Source Offset
6) Currently Linked Media File

Source identifier (4) identifies a specific portion of an original source. The source offset (5) identifies a starting position within the source. The selected media digitization of that source or the currently linked media data file (6) is represented, for example, by a pointer or file name.

To be independent of data type and data sample rate, source references such as the source offset are measured in abstract units called edit units (EU). An EU is a unit of duration, representing the smallest interval of time that is recognized by a given part of a composition, and the number of EU's per second is called an edit rate. An editor may choose

the edit rate for a particular component to represent something useful in the application domain. For example, an application that edits audio based on video frame boundaries would most likely choose an edit rate of 29.97 for the audio tracks, whereas an application that creates audio for use on compact disks would most likely use an edit rate of 44100. Another way of thinking of edit rate is a "virtual" sample rate. This virtual sample rate may or may not match the actual sample rate of the digital media.

The length of a source clip is measured in edit units and the target position in the source is denoted in edit units from a point on the source called the source origin. The origin is an arbitrary place in the source which serves as a reference point for measurements. Additional information concerning source clips is described in U.S. Patent Application Serial No. 08/465,812 filed on June 6, 1995 by Michael J. Wissnor entitled METHOD AND APPARATUS FOR REPRESENTING AND EDITING MULTIMEDIA COMPOSITIONS USING REFERENCES TO TRACKS IN THE COMPOSITION TO DEFINE COMPONENTS OF THE COMPOSITION, incorporated by reference.

A source reference 407 may include a text clip object 411 that may be used to represent text-based media. By specifying such source references as text clip objects, special editing behaviors may be applied to those sources. For example, text-based objects may be edited in a text-based editing system or other editing system capable of revising text.

Interactive elements such as markup language elements may be represented by a new class of text clip 411 referred to as an HTML clip 412. It should be understood that other markup languages may be represented by and HTML clip or may have their own concrete object class that inherits from text clip 411. An editing system 309 may associate HTML clips with particular editing functions specific to markup languages. For example, a broadcast of a television show may display a particular actor, and an interactive element that provides a link to the actor's fan club information may be displayed simultaneously on a viewer's screen. A user may activate this link using an appropriate input device such as a mouse, thus displaying the corresponding fan club information. Thus, the appearance or disappearance of a particular interactive object may be linked to a particular segment of the composition.

A track reference 408 is a concrete subclass of a clip. The track reference 408 represents a contiguous section of media extracted from another track within the same composition. A target offset into the referenced track may be computed relative to the

position of the track reference 408 component within the composition. The specific track references by a track reference 408 may be defined by a track number and a scope number.

The scope number identifies a particular enclosing scope (as defined by the track group components), relative to the track reference 408 itself. A specific scope is found by travelling up the composition structure towards the root object using the parent and subcomponent identifier links, and counting each scope encountered. When the scopes equal the scope number, the reference scope has been located.

The track number identifies a specific track within that scope according to the subcomponent list of the track group.

Other implementations may identify scopes differently, for example, by searching for a specific labeled scope, labeled with a number of string. Other implementations may also identify tracks differently; for example, by searching for a specific label, or by using a relative track number, which is added to the referencing tracks own track number within the scope.

As will be described below, a track group defines a scope, and defines tracks, and a track reference 408 identifies a referenced track by identifying a scope and a track within the scope. The track reference 408 object creates a kind of relative link, which has a very useful property: an assembly of components containing track reference 408 may be copied or removed from an enclosing component, and then reinserted somewhere else, and still by definition refer to a track in the new context. This feature provides the ability to interchange parts of compositions with ease.

Although effects may be built out of wholly contained subtracks of information, it is frequently useful to create an effect which references a separate parallel track in a composition. One example might be an audio track which contains an effect which is the echo of another audio track. A track reference 408 may be used in the effect instead of an explicit subtrack, to make the effect on the first track refer to the second track. A track reference 408 may be also used more generally for implementing layered effects in a composition. Cut and paste of effects is possible: a portion of a composition may be extracted and inserted at a different place in the composition or in a completely different composition, while retaining the internal structural relationships of the contained elements.

An arbitrary user-defined property which varies over time may also be represented as a segment in a composition. Certain user-defined properties may vary in time, in a manner similar to a media source such as video. A track may be added to a composition such that it

represents such a property, and user attribute components can be added to that track, each representing a different value for that attribute.

Transitions 402 may be extended to work with attributes. A camera zooming in from a long shot to a close up could be represented by a transition called an Attribute dissolve placed between a long shot clip and a close up clip.

Fig. 5 shows a typical organization of objects in a composition. A composition object 501 is the root object from which other objects are referenced. The composition object 501 may include a sequence object 502 that may include one or more segments 503 and transitions 505 as described above. A composition created using these objects may have an arbitrarily deep hierarchical structure. A composition 501 is a track group having one or more tracks. In this example, there are three. Each track is represented as a sequence 502 in Fig. 5. A sequence 502A includes a number of segments and transitions 503 and 504. Each segment 503 in this sequence may also include any number of components, such as another track group, another sequence, or simply a source clip. It should now be readily apparent that such a combination represents many different possible types of layering of media in a composition.

A user may define an Interactive Programming track that includes one or more interactive elements. The interactive programming track may be a subclass of the track class. This interactive track may define the transmission times and display time for each of the interactive elements. A playback system may communicate to a viewer system using events, such as sending a trigger to the viewer to display an interactive element or remove the element, among other actions. These events may, for example, cause the media playback system 311 to transmit a multicast datagram including trigger event information. Such information may include, for example, a command or type of event and information identifying the interactive element. Some of these events may be interpreted by the playback system 311 itself. For example, a transmit event may cause playback system 311 to transmit an interactive element to the viewer.

Editing functions regarding track references and other data elements described herein are described in U.S. Patent Application Serial No. 08/465,812 filed on June 6, 1995 by Michael J. Wissner entitled METHOD AND APPARATUS FOR REPRESENTING AND EDITING MULTIMEDIA COMPOSITIONS USING REFERENCES TO TRACKS IN THE COMPOSITION TO DEFINE COMPONENTS OF THE COMPOSITION.

It should be understood that the data structures described herein are but one method of implmenting interactive elements. Media data file formats such as the Open Media Framework (OMF) Interchange format and the Advanced Authoring Format (AAF) and their data structures may be used or modified to incorporate the concepts described herein.

Interactive Programming Track Parameters

Fig. 6A shows an example display view of an interface that controls an interactive programming track. The media editing system 309 may display a dialog box that allows an editor of an interactive multimedia presentation to specify settings appropriate to the Interactive Programming track in a particular MediaComposer sequence. This dialog box may appear if the user has added an Interactive track to the sequence in the timeline. The dialog may allow the editor to specify:

- The **bandwidth** to be used for the Interactive Programming track. When used in conjunction with a VBI insertion device , this value may be specified in terms of the number of lines of VBI which will be devoted to the data stream. This value can determine how long each interactive element will take to download and will be used in calculating whether or not a given sequence of elements in the timeline will fit within the time constraints. For example, for an interactive element that is 55Kbytes in size that is required to be transmitted within 0.5 seconds, 11 lines of the VBI would be required because 11 lines of VBI corresponds to approximately 110 kbps data transmission rate. Allowances for overhead and error correction may be included in determining of required bandwidth.

- The **lead-time** to use for interactive elements in the given composition. The value of lead-time may specify how long (in seconds) a particular element is transferred to a viewer before it may be referenced by an event such as a trigger. For example, the editor might want to provide a five-second window to ensure that the necessary interactive elements will be available on the target computer before displaying a main interactive element such as a main HTML file. Alternatively, the media editing system may explicitly determine an amount of transmission time required to download all required elements, such as dependent interactive elements, to the viewer before sending a trigger event.

- Whether to **download dependencies**. The download dependencies option may determine whether or not files referenced by a given interactive element such as by links (e.g., via the HREF keyword of HTML) are considered to be dependencies of the referring element and thus the dependant elements are automatically downloaded with the referring element. If this option is disabled, only elements explicitly referenced by the main element (such as GIF image files referenced with the IMG keyword of HTML) are downloaded.

Media editing system 309 includes various graphical interfaces that provide an easy-to-use interface for a user to create compositions. Such an interface system may include a system such as the one described in U.S. Application Serial No. 08/689,577 filed on August 12, 1996 by Phillip R. Moorby et al. Entitled MULTIMEDIA VISUAL LANGUAGE METHOD AND APPARATUS. In such a graphical user interface, an icon or other similar symbol may represent a given amount of information or object which may be video data, textual data or audio data or a particular file or instruction. This icon representation may be moved about on the screen by a user and interconnected or linked with other icons to form a multimedia story.

Thus, without having to actually manipulate sound or video materials, the graphical user interface provides a representation of each of those pieces of information which may be manipulated by the user in a fashion he or she desires on the screen, using a mouse or another cursor pointing device.

Once the user has linked the various materials or objects in the desired format, the information may be retrieved in the order established by the user or, in the case of a multimedia presentation, the multimedia presentation's "story" can be "played." This graphical user interface has the advantage that even an unsophisticated user may author somewhat complex presentations without knowing much about computers by simply taking a icon, symbol or other representation of a particular type and given amount of data and manipulating that image on the user's computer screen. The underlying program possesses the intelligence to react to the movement of the representation or symbol on the user's computer screen and manipulate the underlying data which the symbol represents on the user's computer screen. The data structure underlying the operation of the present invention

may be a series of conventional data structure links well known to those skilled in the art which link various pieces of multimedia material.

Thus, from the user's point of view, the movement and interaction of data may be accomplished simply by connecting and interacting icons on the user's screen. The media editing system 309 provides for the capture of a certain amount and type of data as designed by the user or another, such as a video clip or music, and the association of that particular data set with a particular icon representation or symbol.

Media editing system 309 may include a number of graphical components for composing the structure of a story. One is called a timeline. In a timeline, icons representing the story's contents are scaled to depict the length of the presentation of the material for that icon and are placed in proximity to other icons to show timing, synchronization and composition relationships between icons.

A timeline is a simple, time-ordered collection sequences of icons or objects. When a sequence of objects is played, each of the objects is played in order, left to right, top to bottom if there is more than one row of icons. It is to be understood that references to a timeline or sequence means the graphical representation of a series of events which together make up a multimedia presentation.

Most presentations contain graphics, video, and audio material, that may be placed along a timeline. Icons or objects representing this material may be "placed" on the timeline by selecting the object and moving the object to an appropriate point on the timeline. This placing procedure is commonly referred to in the art as "drag-and-drop." In a timeline, icons (such as icons that represent media perform action, and link with other icons) are placed along the timeline to represent the contents (such as graphics, video, and audio, etc.) of the presentation. Because the computer can store information in many ways, icons can store presentation content by utilizing many different kinds of computer-based file formats. Data may be stored in a file during editing and playing. In addition, if desired, data may be stored in an external storage medium or computer. A timeline, once completed or partially completed, may be played using the media editing system 309. The position of the icons along the timeline determine the sequence by which the presentation is played back to the viewer. An example of a timeline is shown in Fig. 6C.

Importing Elements into Bins

Modifying a conventional media editing system 309 editing system to handle interactive elements offers many advantages. Fig. 6B shows a media composer 309 display view showing a media editing system "bin" that lists available interactive elements that may be included in a composition. The ability to import Interactive Programming elements such as HTML and other file types into standard media editing system bins allows the editing system to operate on them in much the same way as other media elements (organizationally, etc.). To perform the import, a user may select the bin and choose an "import" command from the appropriate menu. The file type of Interactive Programming is chosen and the dialog box displays all applicable element files. The user then may select one or more files and select an appropriate button to perform the import of the interactive element files. The editing system 309 then may list the interactive elements along with the files that contain them.

When an Interactive Programming element is imported into a media editing system bin, the element file is scanned to see what other element files, if any, are referenced by it (for example, if there are image files referenced by the main HTML file). These element files may be automatically imported together with the main file. When the user specifies that a particular HTML file is to be used in the sequence, these files may be included implicitly as well. Just as with other media elements in the bin, the user can select an Interactive Programming element to obtain information about it through the media editing system 309 interface. This information may include the location of the original file. If the user edits the original file, he or she may re-import the file in order to update the copy within the media editing system bin.

Interactive Programming Track in Timeline

When the user chooses the "New Interactive Programming Track" command in the Clip menu, an Interactive Programming track is added to the sequence currently displayed in the Timeline window. Fig. 6C shows a media editing system 309 display view including an Interactive Programming Track IP1. This interactive programming track allows the placement of Interactive Programming elements and events into the sequence being edited, through the drag-and-drop metaphor described above with respect to the graphical user interface of the media editing system 309.

Element Appearance in Timeline

Interactive Programming elements appear in the timeline in much the same way as other media elements such as video elements. The position and size of the object in the timeline reflect the time during which the object will appear on the target viewer's display. For example, referring to Fig. 6C, an interactive object "SportsCenter Main Page" begins at approximately time 01:00:05:00, and thus will appear at that time on the viewer's display. Also, the same interactive object ends at approximately 01:00:05:50, and thus the object will cease to be displayed at that time. Note that the interactive object "SportsCenter Main Page" begins at the same point in the timeline as the "Hockey Game 01" video element in the video track V1. Thus, the "SportsCenter Main Page" and "Hockey Game 01" elents are associated with each other through the timeline, and may be displayed at the same time. As described below, although the interactive element "SportsCenter Main Page" is displayed at the same time as the "Hockey Game 01" video element, the interactive element may need to be transmitted by the playback system at an earlier time than the transmission of the video element.

Enhancement Events in Timeline

Other than Interactive Programming elements themselves (such as HTML and GIF files), the user will add events to the timeline. Some events, such as triggers to display a given HTML element on the target machine are generated implicitly by adding the element to the timeline. For example, in the case of displaying the "SportsCenter Main Page" object shown in Fig. 6C, media editing system 309 may indicate that a display event should be transmitted to a viewer at the 01:00:04:00 time in the timeline. Similarly, an "end display" event may be sent to remove the "SportsCenter Main Page" object at time 01:00:05:50. Other events, such as File Transfers, Ticker Data, Chat Data, Messages, and custom events may be added to display different types of interactive elements.

Workflow

The following description demonstrates how an editor would add interactive objects to an already edited sequence within the media editing system 309. The edition of the interactive object track may occur in parallel with the edition of the video and audio tracks. This description, however, describes sequential editing to more clearly illustrate the workflow

issues brought into play by the addition of the Interactive Programming features. It is understood that the editor may use different editing sequences to achieve a same result.

Interactive Content Creation

The creation and editing of the interactive elements in a composition may be prepared in advance of their inclusion in the timeline sequence by using an appropriate tool such as Microsoft's FrontPage HTML editing system application (FrontPage is a registered trademark of the Microsoft Corporation). These files may be created on a system on which the media editing system 309 resides, have been copied to the media editing system 309, or reside in a location accessible to the media editor 309 system over the network, such as on an Internet server. Media editing system 309 may be modified to edit these interactive elements directly.

Importing Interactive Programming Elements

To work with the Interactive Programming elements, the editing system may import them into a bin within the target project. Importing is accomplished in the same way as any other import, except that the user is now prompted to locate the interactive element files. When importing the files, media editing system 309 may maintain a link to the original files on the user's hard drive (or networked drive) or other appropriate storage area. If the files change at any point in the future, the user may be notified and may have the option of updating the version stored in the bin. The editor may use the bins in his or her project to organize Interactive Programming elements in a conventional way as is done with audio and video elements. Double-clicking the object in the bin opens the element in an HTML or other editing system as appropriate. The editing system may be defined by the user's settings.

When importing an HTML file, the media editing system 309 may automatically import any referenced files as appropriate. These referenced files can appear as reference clips within the bin (similarly to the appearance of objects referenced by a sequence, but not explicitly added to the bin themselves). Depending on the user's preferences, files referenced through links in the imported HTML file may or may not be included as reference clips.

The Timeline's Interactive Programming Track

Once the desired clips are available in the bin, the user can use a menu command to add an Interactive Programming track to the timeline containing the sequence he or she is

editing. The user, through the editing system interface will be able to drag previously imported elements into the track or may open them in the source monitor and then splice or overwrite them into the timeline.

Adding Interactive Elements to the Timeline

After the principal interactive element is in place, a user may operate the editing system to add whatever other elements are appropriate at the proper points in the timeline. Interactive Programming elements placed into the timeline appear similar to the way other media elements do. They have a length which indicates the duration for which they will be displayed on the user's screen. The editor may operate with the interactive element data in the same way as he or she would use any other media in the timeline. The editor may be alerted if there is not enough time before an interactive element is to be shown for it to be downloaded. If there is not enough time, the user can either move the interactive element further down the timeline or remove non-critical elements ahead of the element in the timeline.

Inserting Interactive Programming Events

These trigger events may be generated automatically by inserting an element into the timeline. Other types of events may be inserted using standard video effects within the media editing system 309 to provide things such as messages, file transfers, etc.

Previewing

A user may preview the interactive elements within an interface of the media editing system 309. When the timeline sequence is played in the media editing system 309, the Interactive Programming track can be played as well, resulting in output to a VBI insertion device. The editor may preview the output video presentation using a WebTV box connected to a standard television monitor or using a second PC with the appropriate hardware running Windows 98's TV viewer. Other systems may be used to preview the presentation.

Output

Output of Interactive Programming track is output in a manner similar to other media output. The user chooses a digital cut command and editing system 309 may output the data directly to a medium such as videotape. As the sequence is being played, the Interactive

Programming elements may be simultaneously output via a serial or other type connection to the VBI encoder. As discussed above, the VBI encoder takes the video output from the media editing system 309 and inserts the enhancement information into it, producing an enhanced video stream. The enhancement information may include markup language elements. This information can be stored on standard videotape, stored in some digital media format, or broadcast directly from the media editing system 309.

Example Edit

Figure 7 describes a process for creating a composition including interactive elements. A user, through the media editing system 309, opens a bin of video elements at step 701. At step 703, the user creates a short sequence using pre-defined edit points. At step 705, the user positions the cursor on different shots in the sequence. For example, the user would like to associate information regarding the star who is currently displayed during the short sequence. Another example could include placing a link to a downloadable file including the script from the video. Many other associations between interactive elements and the video presentation are possible.

At step 707, the user positions the cursor over one shot and marks the clip. This marks the entire range (duration) of the shot and creates a new timeline track. At step 709, the user invokes the Interactive Programming track. This track is used to facilitate the encoding of embedded information for Interactive Programming of interactive elements. At step 711, the user creates an Interactive Elements bin and imports the Interactive Programming elements that will be used within the composition. At step 713, the user patches the Interactive Programming characteristic (track) of the source element to the Interactive Programming track in a well-known manner. The user may continue to mark clips further in the timeline, call up new interactive elements, and overwrite them into position. At step 715, the user may open the Interactive Programming Parameters dialog box to adjust bandwidth, lead-time, and dependencies for the transmission of the interactive elements. As discussed above, the Interactive Programming parameters will encode and transmit the interactive elements in real time through the VBI the information that has just been added to the sequence via the Interactive Programming track in the timeline. At step 717, the user may play back the composition and watches the encoded results on a viewer such as a WebTV-based viewer. The user may further add interactive elements to the composition and review the composition

in a recursive manner. It should be understood that many other editing sequences are possible, and that this is merely one example of an editing process.

Determining Bandwidth Requirements

Figure 8 describes a process for determining transmission bandwidth requirement for transmitting an interactive element. At step 801, editing system 309 determines the transmission requirements for transmitting the interactive element. At step 803, editing system 309 determines whether requirements can be met by the current data transmission channel. If the data transmission is a VBI channel, the amount of transmission bandwidth will depend on the amount of lines of VBI used to transmit the interactive element. If these bandwidth transmission requirements cannot be met, the editing system 309 may indicate an error to the user at step 805. If the bandwidth requirements can be met, editing system 309 schedules a time for transmission of the interactive elements at step 807. The editing system 309 may develop a transmit event associated with the transmission of the element. This transmit event may be interpreted by the media playback system 311 as a signal to being transmission of the interactive element. At step 820, the determination process ends.

Associating Interactive Elements with Video Track Elements

Figure 9 shows a process for associating interactive track elements with video track elements. At step 901, a video track is created or opened at editing system 309. At step 903, a user may create an interactive track using editing system 309. At step 905, the editing system 309 associates an interactive track element with a video element. This association may be an explicit link to the video element from the interactive track element, or a relation of the video element to the interactive track element by their positions in the timeline. At step 906, the associating process ends.

Transmitting Interactive and Video Track Information

Figure 10 shows a process for transmitting interactive and video track information. At step 1001, playback system 311 obtains an interactive element identified in the interactive track. This interactive track may have been defined previously using editing system 309. The interactive element may be, for example, stored as a file on storage system 209, or any other storage system throughout the network. This interactive track element may be placed in an

interactive track buffer for transmission. Similarly, playback system 311 may obtain a video track element and place the element in a video track buffer for transmission at step 1003. At step 1005, playback system 311 may sequence the transmission of the interactive track and video elements. As discussed above, on or more of the interactive elements may need to be transmitted to the viewer before transmitting video track elements associated with the interactive elements. At block 1007, the interactive elements and video elements are transmitted. As discussed above, video elements may be transmitted as video data such as a video broadcast signal (e.g. NTSC signal) and the interactive elements may be transmitted as encoded data within the VBI of the video broadcast signal. Other methods of transmitting the video and interactive data are possible.

Having now described a few embodiments of the invention, it should be apparent to those skilled in the art that the foregoing is merely illustrative and not limiting, having been presented by way of example only. Numerous modifications and other embodiments are within the scope of one of ordinary skill in the art and are contemplated as falling within the scope of the invention as defined by the appended claims and equivalents thereto.

CLAIMS

1. A method for processing a video signal, the method comprising steps of:

associating an interactive element to a portion of a video presentation to be transmitted and displayed on a viewer, wherein the interactive element has a predefined relationship to the portion; and

transmitting the interactive element to the viewer at a predetermined time relative to a transmission of the portion.

2. A method for distributing a video signal produced by the method of claim 1.

3. A method for storing a video signal produced by the method of claim 1.

4. The method according to claim 1, wherein the step of associating includes a step of associating an interactive element to a video element by placing both the interactive element and the video element at a same starting position in a timeline.

5. The method according to claim 1, wherein the predetermined time is a time before the transmission of the video element to ensure the interactive element is available for display at the viewer at a time that the video element is viewed.

6. A system for transmitting a video signal comprising:

means for determining a time relation between a video broadcast and an interactive video element; and

means for inserting, into the video signal, the interactive element based on the time relation.

7. A method for editing a video signal comprising steps of:

determining a time relation between a video broadcast and an interactive video element; and

inserting, into the video signal, the interactive element based on the time relation.

8. A method for transmitting a video signal comprising steps of:

receiving a composition of a multimedia presentation, the composition including one or more interactive elements having a time relation to one or more video elements;

encoding the interactive elements within a video signal; and

transmitting the video signal.

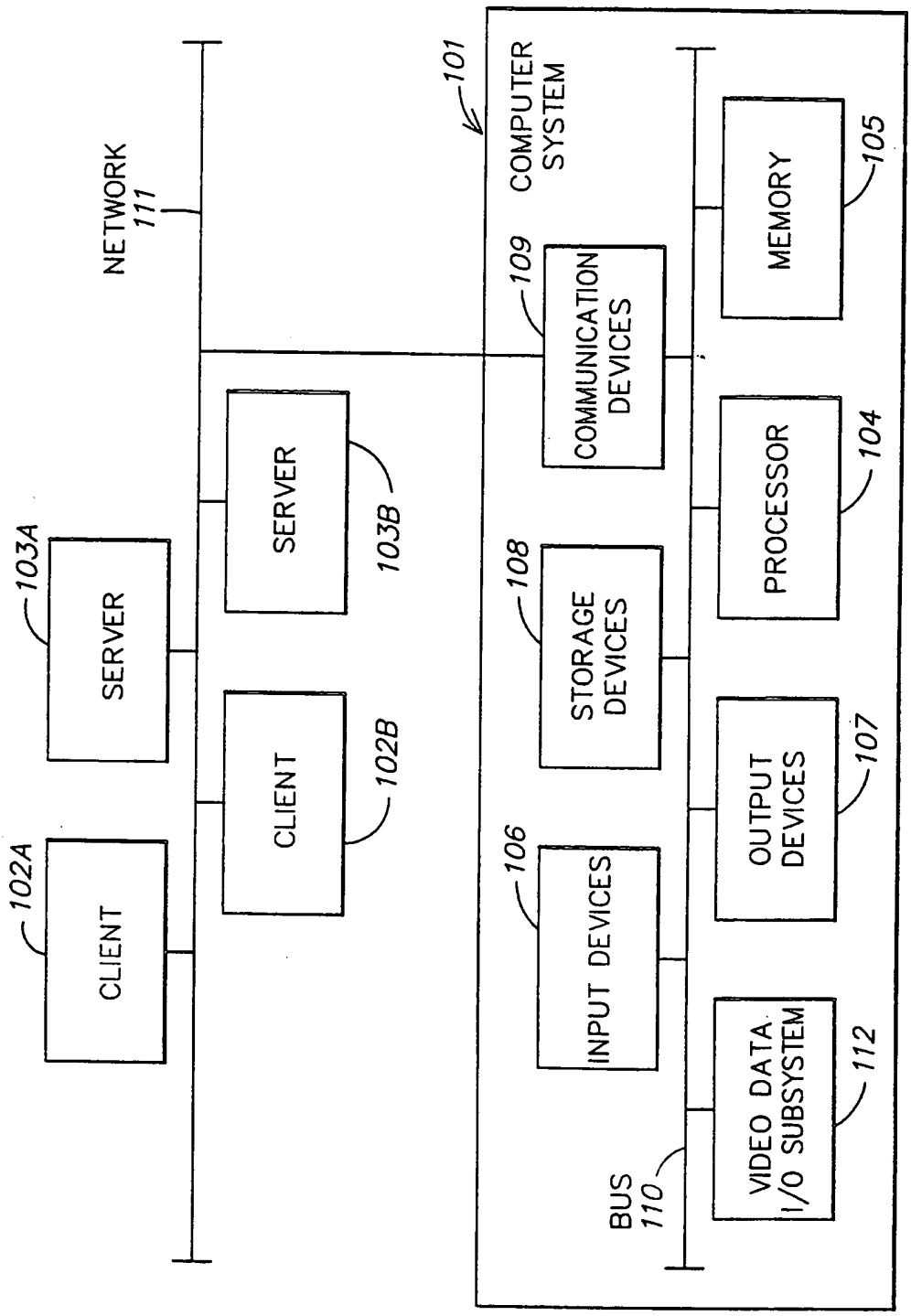

NETWORK
111
101
COMPUTER SYSTEM
109
COMMUNICATION DEVICES
MEMORY
105
SERVER
103B
PROCESSOR
104
103A
SERVER
108
STORAGE DEVICES
CLIENT
102B
OUTPUT DEVICES
107
102A
CLIENT
106
INPUT DEVICES
VIDEO DATA I/O SUBSYSTEM
112
BUS
110

FIG. 1

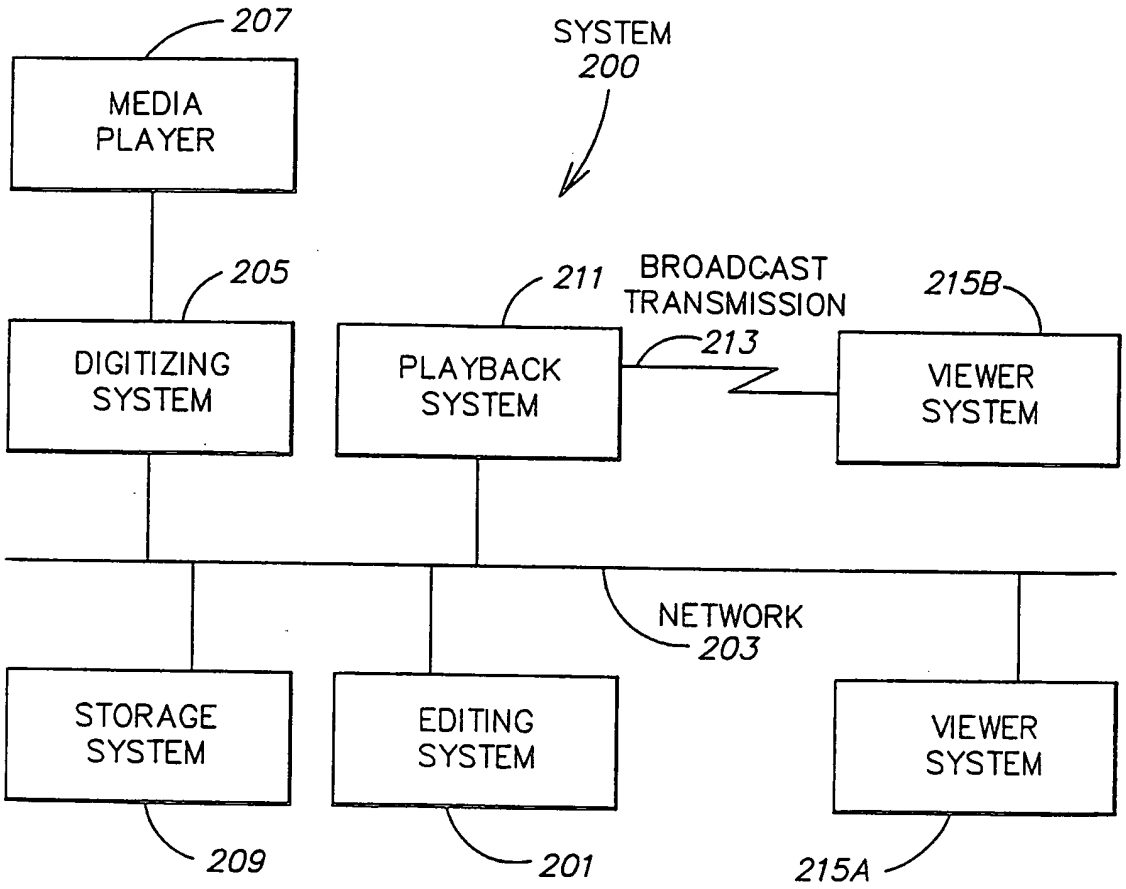

207
MEDIA PLAYER
SYSTEM
200
205
DIGITIZING SYSTEM
211
BROADCAST TRANSMISSION
213
PLAYBACK SYSTEM
215B
VIEWER SYSTEM
NETWORK
203
STORAGE SYSTEM
EDITING SYSTEM
VIEWER SYSTEM
209
201
215A

FIG. 2

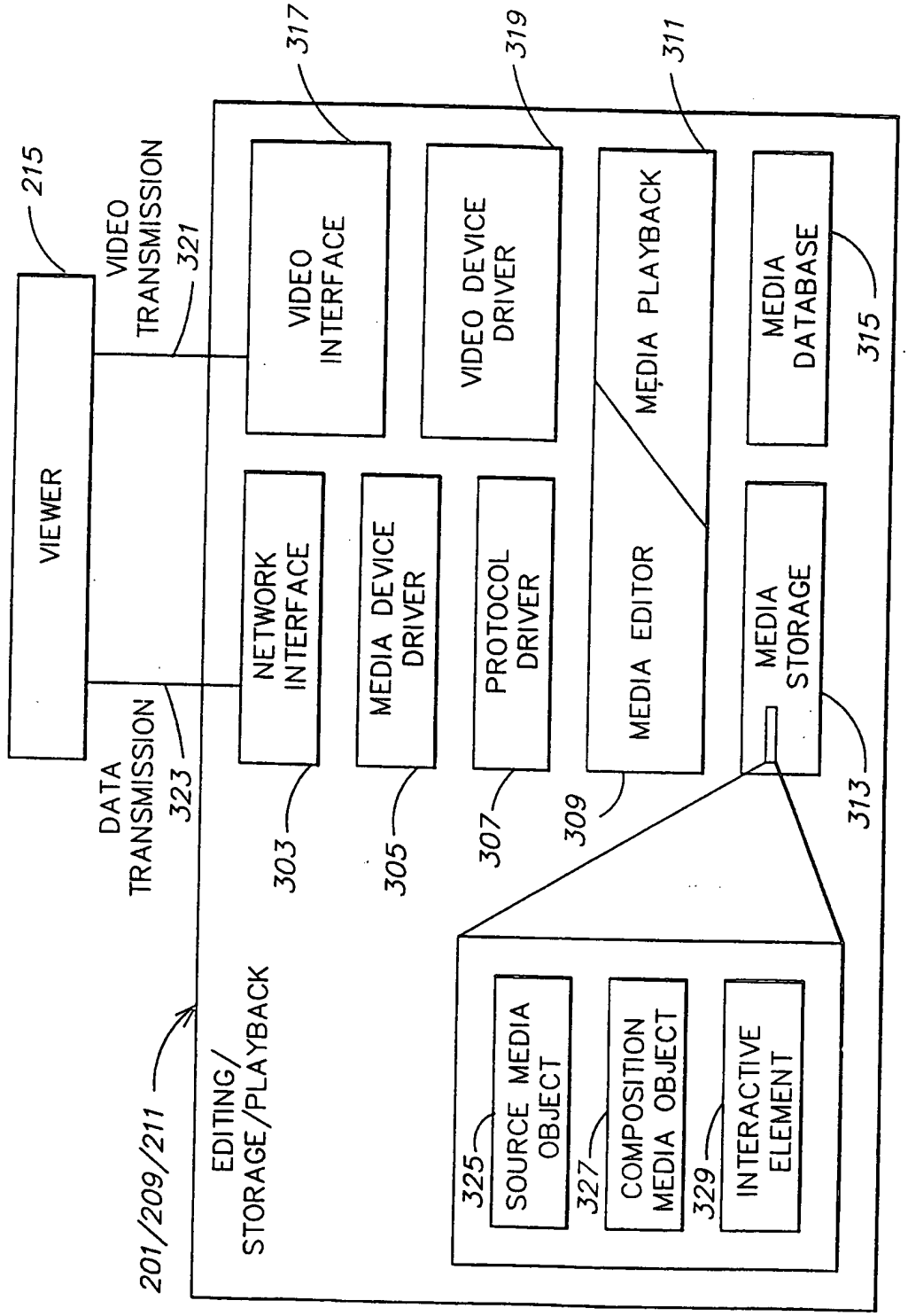

215
VIEWER
VIDEO TRANSMISSION
321
DATA TRANSMISSION
323
317
VIDEO INTERFACE
319
VIDEO DEVICE DRIVER
311
MEDIA PLAYBACK
MEDIA EDITOR
315
MEDIA DATABASE
NETWORK INTERFACE
303
MEDIA DEVICE DRIVER
305
PROTOCOL DRIVER
307
309
MEDIA STORAGE
313
EDITING/ STORAGE/PLAYBACK
201/209/211
325
SOURCE MEDIA OBJECT
327
COMPOSITION MEDIA OBJECT
329
INTERACTIVE ELEMENT

FIG. 3A

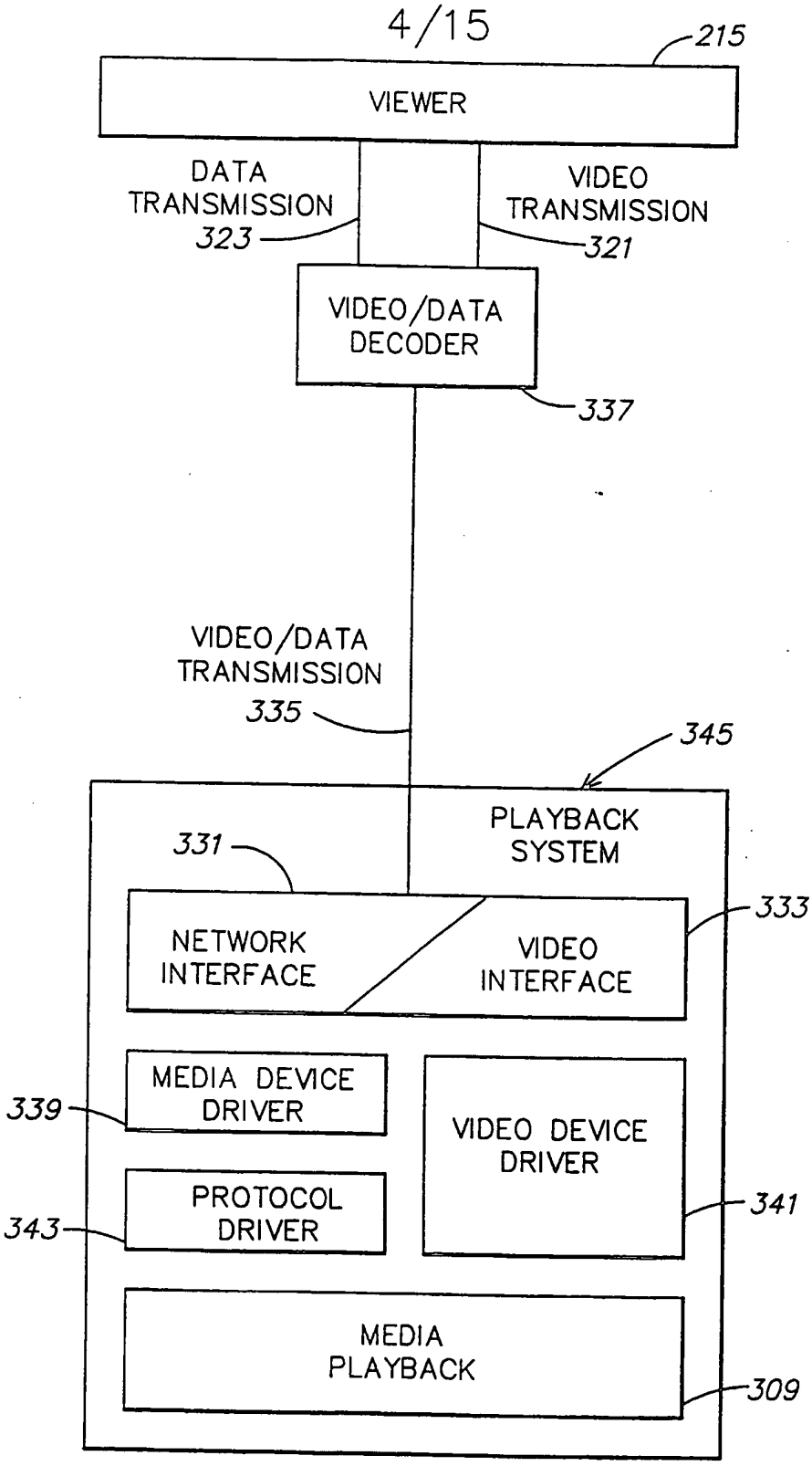

215
VIEWER
DATA TRANSMISSION
323
VIDEO TRANSMISSION
321
VIDEO/DATA DECODER
337
VIDEO/DATA TRANSMISSION
335
345
PLAYBACK SYSTEM
331
333
NETWORK INTERFACE
VIDEO INTERFACE
MEDIA DEVICE DRIVER
339
VIDEO DEVICE DRIVER
341
PROTOCOL DRIVER
343
MEDIA PLAYBACK
309

FIG. 3B

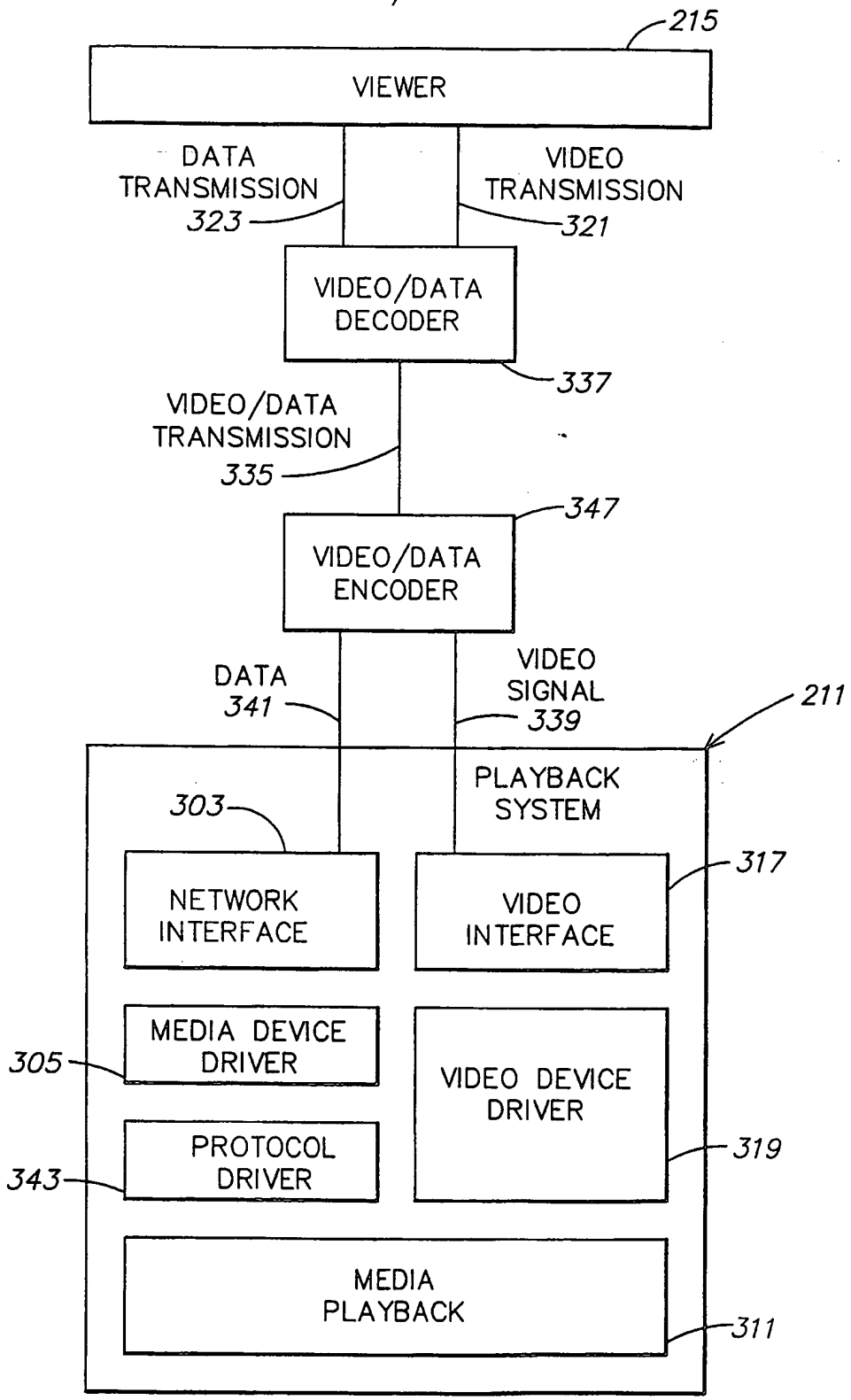

215
VIEWER
DATA TRANSMISSION
323
VIDEO TRANSMISSION
321
VIDEO/DATA DECODER
337
VIDEO/DATA TRANSMISSION
335
347
VIDEO/DATA ENCODER
DATA
341
VIDEO SIGNAL
339
211
PLAYBACK SYSTEM
303
NETWORK INTERFACE
VIDEO INTERFACE
317
MEDIA DEVICE DRIVER
305
VIDEO DEVICE DRIVER
PROTOCOL DRIVER
343
319
MEDIA PLAYBACK
311

FIG. 3C

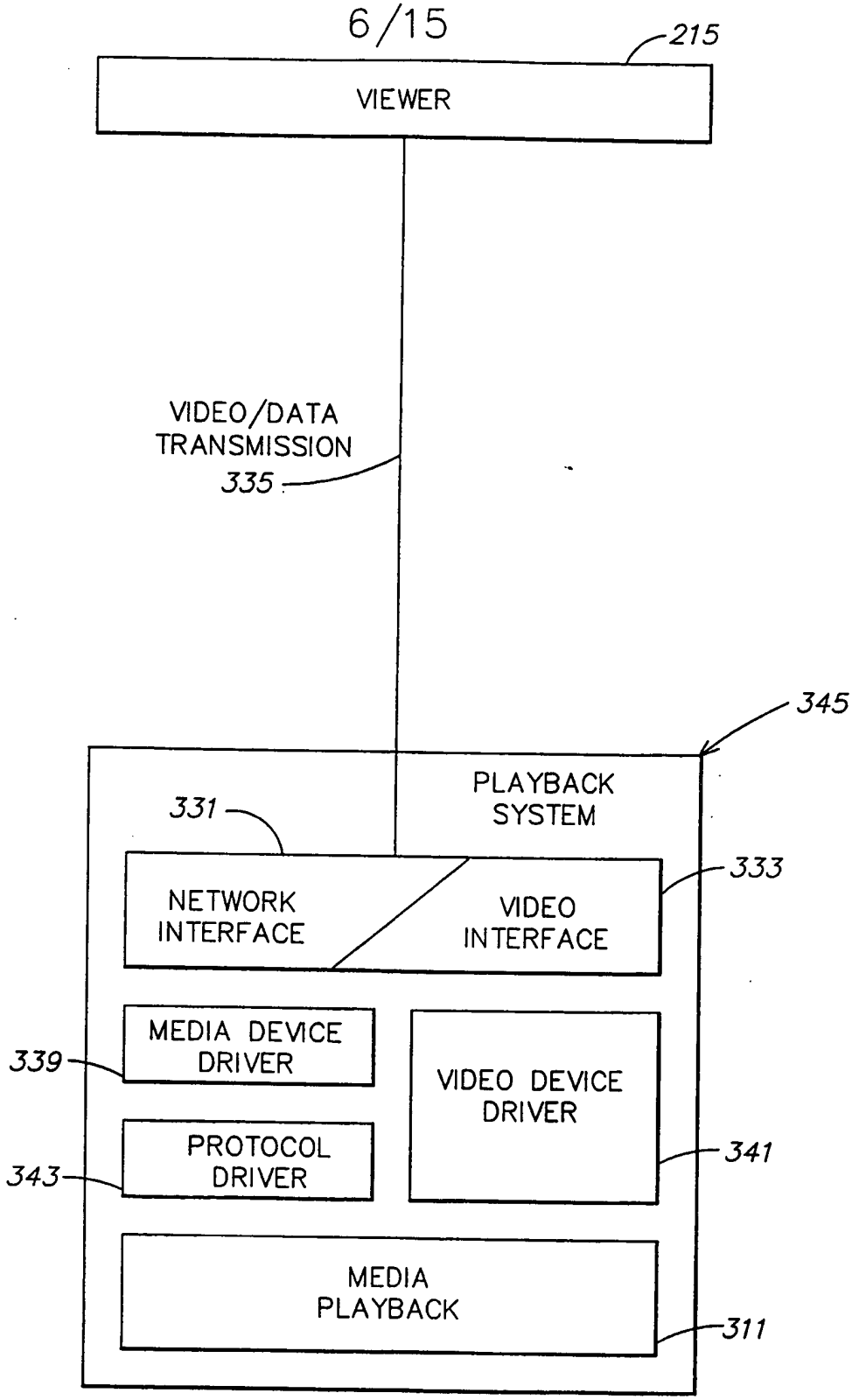

215
VIEWER
VIDEO/DATA
TRANSMISSION
335
345
PLAYBACK
SYSTEM
331
333
NETWORK
INTERFACE
VIDEO
INTERFACE
MEDIA DEVICE
DRIVER
339
VIDEO DEVICE
DRIVER
PROTOCOL
DRIVER
343
341
MEDIA
PLAYBACK
311

FIG. 3D

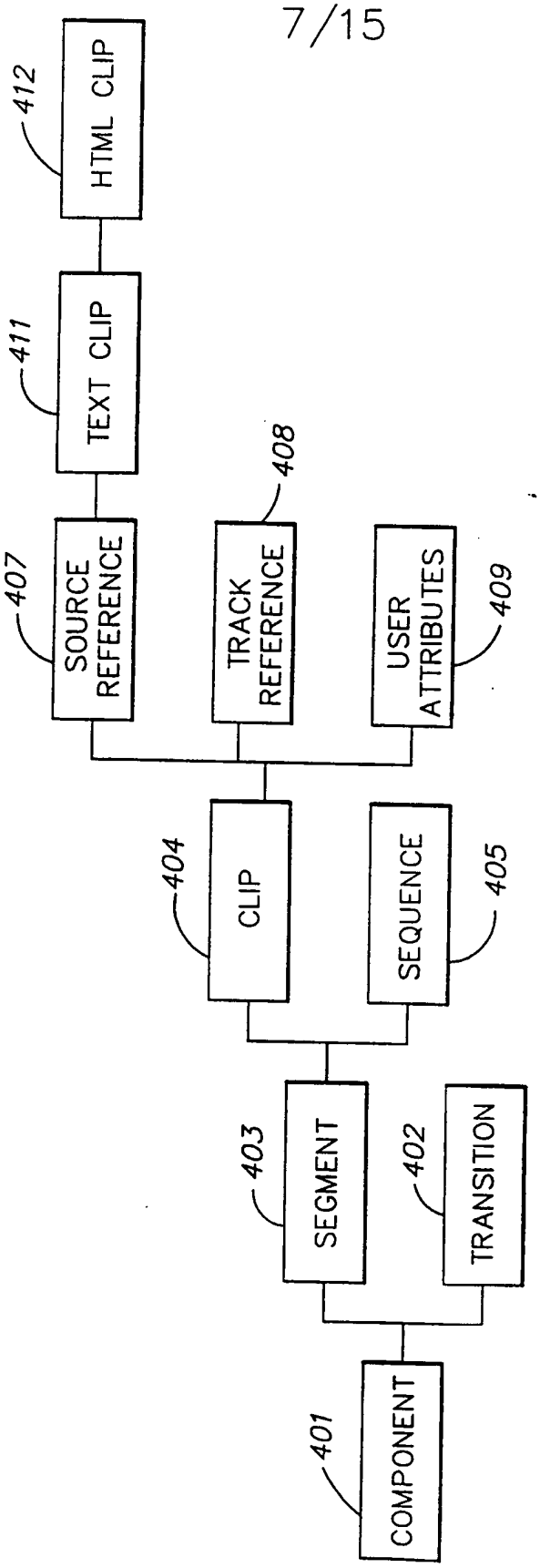

412
HTML CLIP
411
TEXT CLIP
407
SOURCE REFERENCE
408
TRACK REFERENCE
409
USER ATTRIBUTES
404
CLIP
405
SEQUENCE
403
SEGMENT
402
TRANSITION
401
COMPONENT

FIG. 4

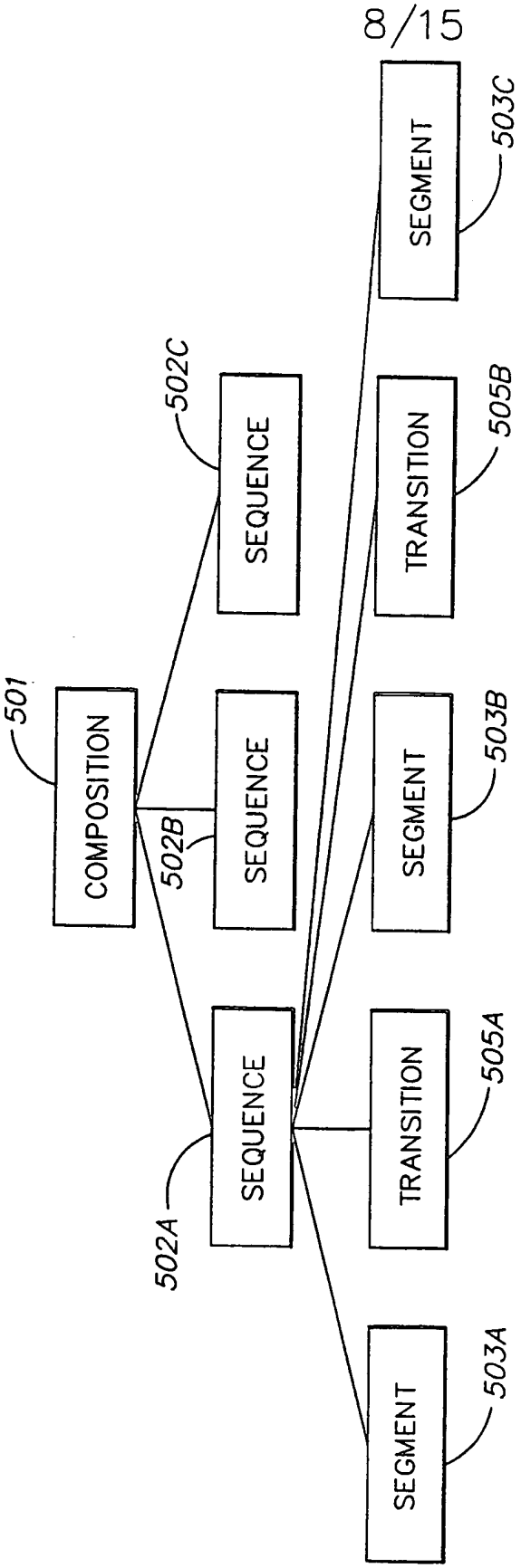

501
COMPOSITION
502A
SEQUENCE
502B
SEQUENCE
502C
SEQUENCE
503A
SEGMENT
503B
SEGMENT
503C
SEGMENT
505A
TRANSITION
505B
TRANSITION

FIG. 5

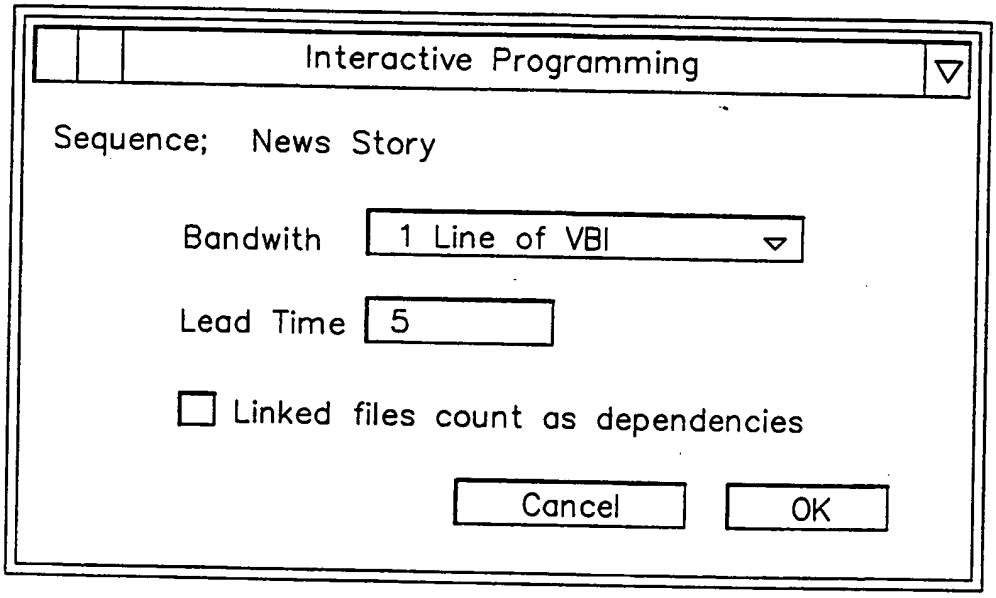

Interactive Programming
Sequence; News Story
Bandwith
1 Line of VBI
Lead Time
5
Linked files count as dependencies
Cancel
OK

FIG. 6A

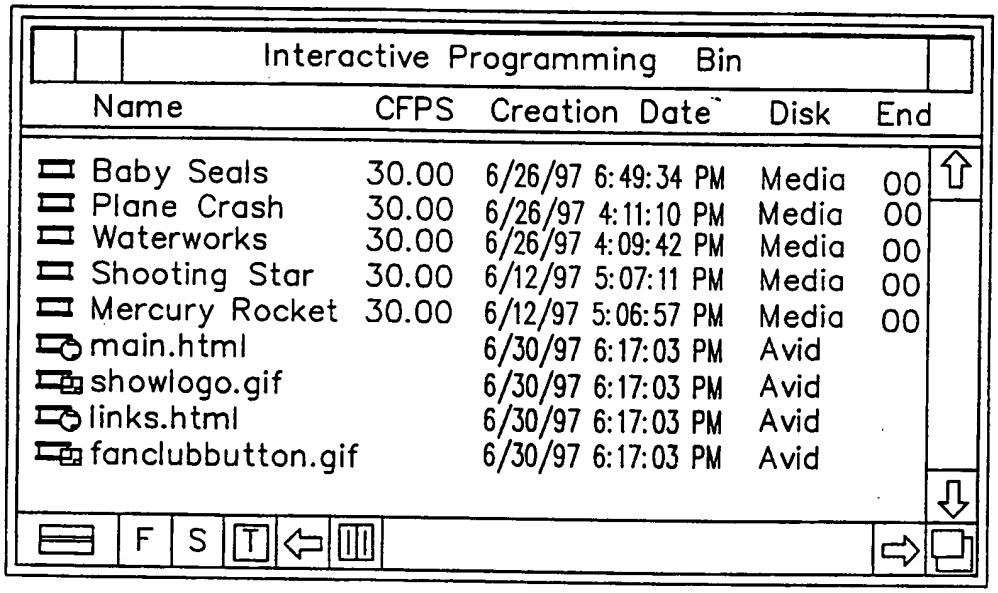
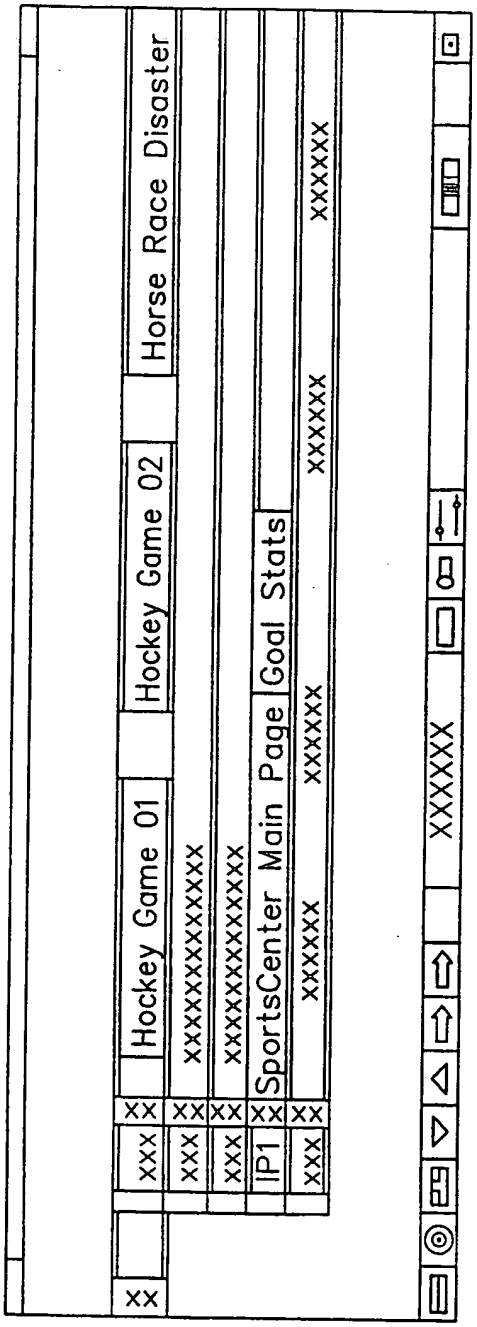

Interactive Programming Bin
Name
CFPS
Creation Date
Disk
End
Baby Seals 30.00 6/26/97 6:49:34 PM Media 00
Plane Crash 30.00 6/26/97 4:11:10 PM Media 00
Waterworks 30.00 6/26/97 4:09:42 PM Media 00
Shooting Star 30.00 6/12/97 5:07:11 PM Media 00
Mercury Rocket 30.00 6/12/97 5:06:57 PM Media 00
main.html 6/30/97 6:17:03 PM Avid
showlogo.gif 6/30/97 6:17:03 PM Avid
links.html 6/30/97 6:17:03 PM Avid
fanclubbutton.gif 6/30/97 6:17:03 PM Avid
F
S
T

FIG. 6B


Hockey Game 01
Hockey Game 02
Horse Race Disaster
SportsCenter Main Page
Goal Stats
xxxxxxxxxxxx
xxxxxxxxxxxx
xxxxxx
xxxxxx
x

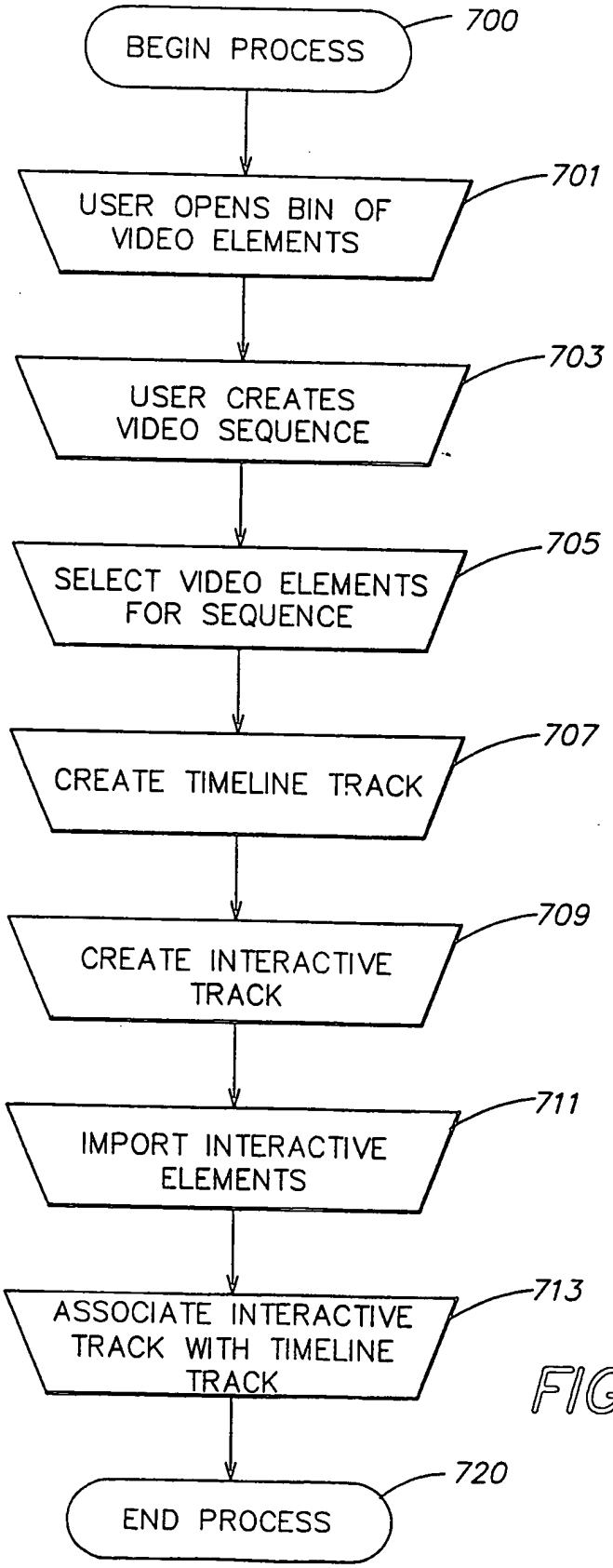

BEGIN PROCESS
700
USER OPENS BIN OF VIDEO ELEMENTS
701
USER CREATES VIDEO SEQUENCE
703
SELECT VIDEO ELEMENTS FOR SEQUENCE
705
CREATE TIMELINE TRACK
707
CREATE INTERACTIVE TRACK
709
IMPORT INTERACTIVE ELEMENTS
711
ASSOCIATE INTERACTIVE TRACK WITH TIMELINE TRACK
713
END PROCESS
720

FIG. 7

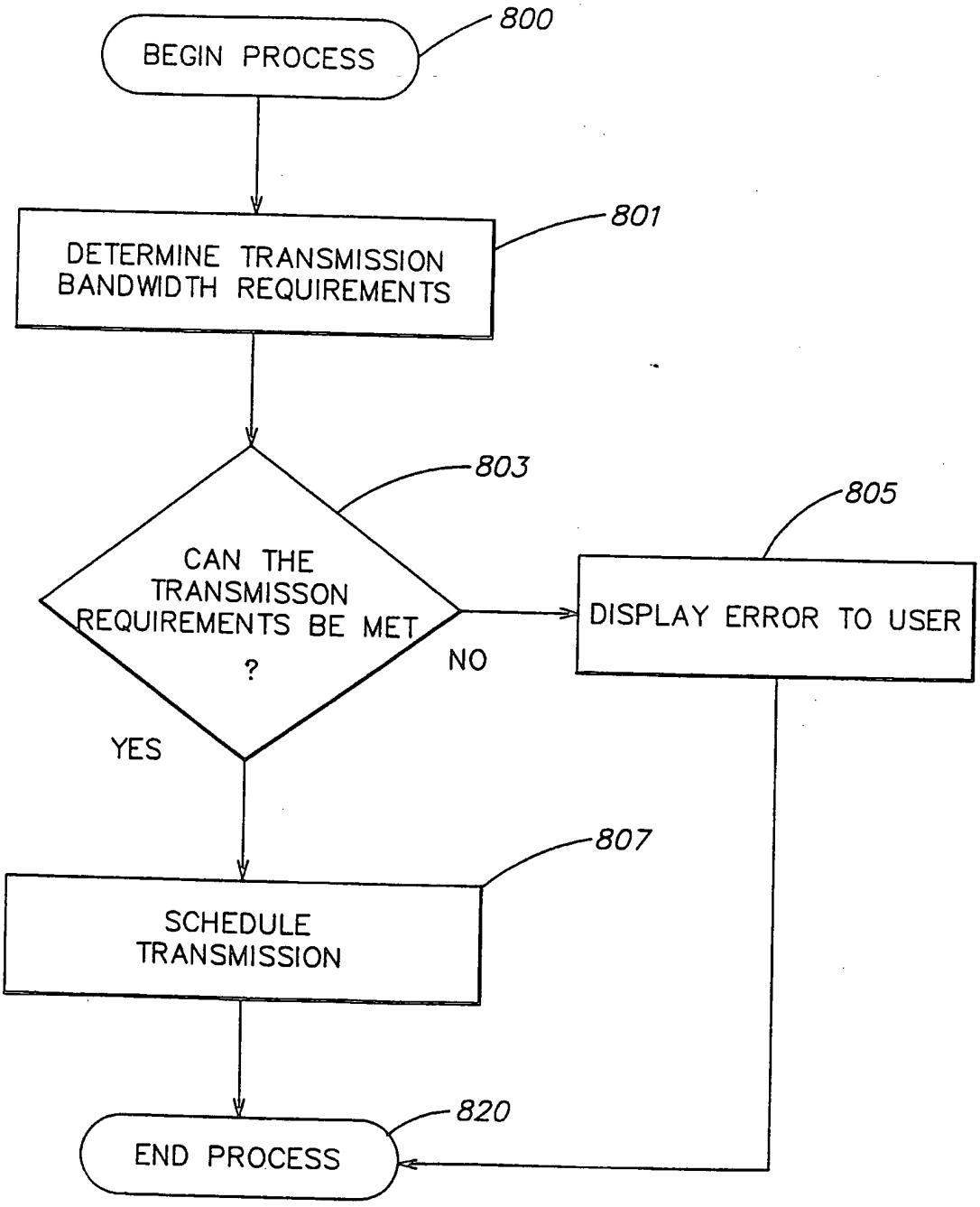

BEGIN PROCESS
800
DETERMINE TRANSMISSION BANDWIDTH REQUIREMENTS
801
CAN THE TRANSMISSON REQUIREMENTS BE MET ?
803
NO
DISPLAY ERROR TO USER
805
YES
SCHEDULE TRANSMISSION
807
END PROCESS
820

FIG. 8

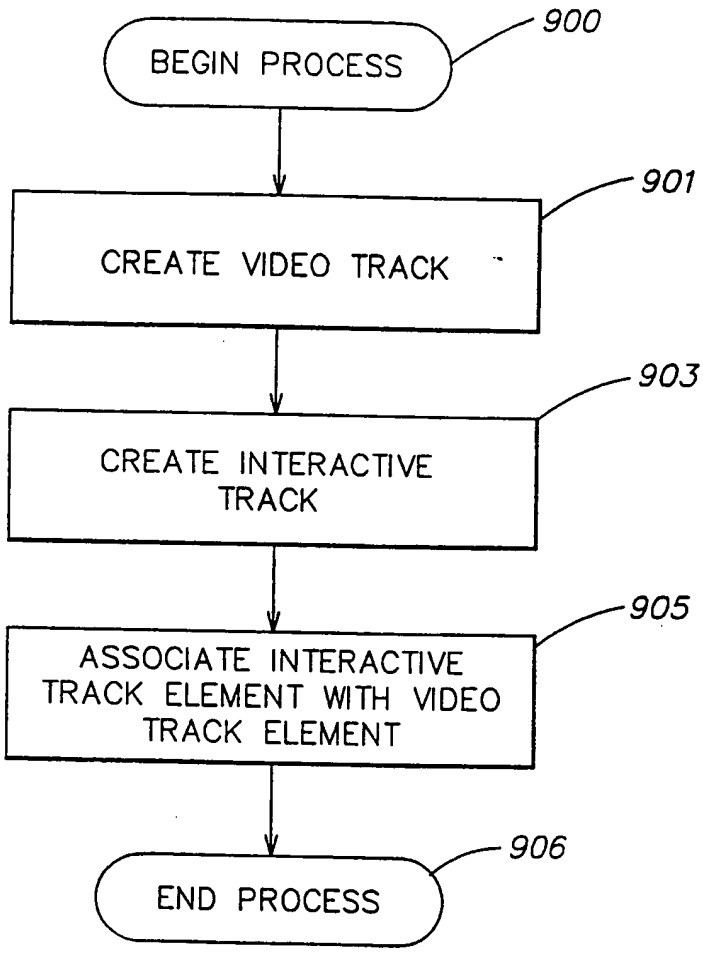

BEGIN PROCESS
900
CREATE VIDEO TRACK
901
CREATE INTERACTIVE TRACK
903
ASSOCIATE INTERACTIVE TRACK ELEMENT WITH VIDEO TRACK ELEMENT
905
END PROCESS
906

FIG. 9

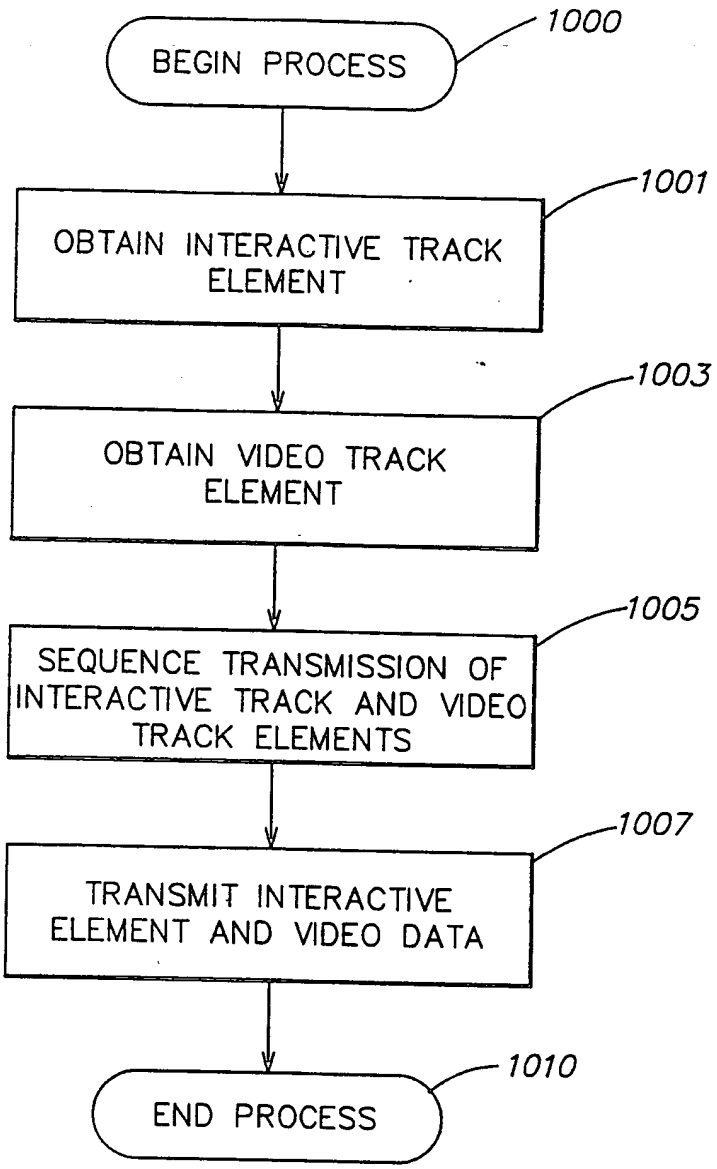

BEGIN PROCESS
1000
OBTAIN INTERACTIVE TRACK ELEMENT
1001
OBTAIN VIDEO TRACK ELEMENT
1003
SEQUENCE TRANSMISSION OF INTERACTIVE TRACK AND VIDEO TRACK ELEMENTS
1005
TRANSMIT INTERACTIVE ELEMENT AND VIDEO DATA
1007
END PROCESS
1010

FIG. 10

INTERNATIONAL SEARCH REPORT

Interr al Application No
PCT/US 99/06724

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 97 37497 A (BRITISH TELECOMM ;ASTIZ PAUL (US); FEIT FIL (US)) 9 October 1997 see abstract; claim 1; figures 1-4 see page 11, line 23 - page 24, line 17	1-8
X	EP 0 596 823 A (IBM) 11 May 1994 see column 4, line 50 - column 5, line 52; figure 3	1-8
X	HUNG Y -C: "A HYPERVIDEO SYSTEM GENERATOR" SOFTWARE PRACTICE & EXPERIENCE, vol. 27, no. 11, 1 November 1997, pages 1263-1281, XP000703783 see page 1268, line 31 - page 1276, line 35; figures 2-10	1-8

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
2 July 1999	09/07/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Deane, E

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Intern al Application No
PCT/US 99/06724

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 97 12342 A (WISTENDAHL DOUGLASS A ;CHONG LEIGHTON K (US)) 3 April 1997 see page 8, line 4 - page 9, line 15	1-8
X	TONOMURA Y: "VIDEO HANDLING BASED ON STRUCTURED INFORMATION FOR HYPERMEDIA SYSTEMS" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON MULTIMEDIA INFORMATION SYSTEMS,1 January 1991, pages 333-344, XP002004970 see page 336, line 3 - line 33; figures 2,3	1-8
X	WO 98 04984 A (EPHYX TECHNOLOGIES LTD ;EPHYX TECHNOLOGIES INC (US)) 5 February 1998 see page 23, line 10 - page 25, line 8; figure 9A	1-8
A	HIRATA K ET AL: "CONTENT-ORIENTED INTEGRATION IN HYPERMEDIA SYSTEMS" HYPERTEXT '96. 7TH. ACM CONFERENCE ON HYPERTEXT, WASHINGTON, MAR. 16 - 20, 1996, no. CONF. 7, 16 March 1996, pages 11-21, XP000724319 ASSOCIATION FOR COMPUTING MACHINERY see page 18, left-hand column, line 4 - line 31; figure 7	1-8
A	WO 96 36007 A (MAGIC CIRCLE MEDIA INC) 14 November 1996 see abstract; figures 1-3 see page 6, line 8 - page 7, line 21	1-8

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Intern: al Application No
PCT/US 99/06724

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9737497 A	09-10-1997	AU 2333397 A	22-10-1997
		EP 0891675 A	20-01-1999
EP 0596823 A	11-05-1994	US 5539871 A	23-07-1996
		JP 2677754 B	17-11-1997
		JP 7085243 A	31-03-1995
WO 9712342 A	03-04-1997	US 5708845 A	13-01-1998
		CA 2233444 A	03-04-1997
		EP 0902928 A	24-03-1999
WO 9804984 A	05-02-1998	AU 3894697 A	20-02-1998
		EP 0922259 A	16-06-1999
WO 9636007 A	14-11-1996	AU 5735696 A	29-11-1996
		CA 2220277 A	14-11-1996
		CN 1185850 A	24-06-1998
		EP 0870273 A	14-10-1998